Motoman ERC Controller

Communications Manual

Part Number: 479236-17
Revision: 2

Motoman, Incorporated
805 Liberty Lane
West Carrollton, OH 45449
TEL: (937) 847-6200
FAX: (937) 847-6277
24-Hour Service Hotline: (937) 847-3200

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 WHO SHOULD READ THIS DOCUMENTATION

YASNAC ERC CONTROLLER COMMUNICATIONS is a document written for engineers and/or programmers with prior data communications and Yasnac ERC Controller experience. The document is intended to supplement the user's guides and operator's manuals that are provided with a Motoman system. YASNAC ERC CONTROLLER COMMUNICATIONS can be used as an implementation guide and reference tool for communications between the Yasnac ERC Controller and an external device.

1.2 HOW TO USE THIS DOCUMENT

YASNAC ERC CONTROLLER COMMUNICATIONS can be used as a reference document when implementing communications between the Yasnac ERC Controller and an external device. It is not necessary for the user to read this document from beginning to end. However, the knowledge contained within each section may be useful or provide ideas as to alternative options and features of the ERC Controller.

The user should read Sections 3.0 and 4.0 which provide a basis for the details within the remaining sections. Following these two sections, the user should read the particular section(s) which relate to the communications application to be implemented.

Detailed specifications have been included in each section of this document such that it can be used as an implementation guide. Information such as pinout descriptions, communication parameter settings, command formats, transmission sequence and example transactions can be found in the individual sections.

Following implementation of the communications application, this document should be retained for use as a reference guide.

1.3 STRUCTURE OF THIS DOCUMENT

YASNAC ERC CONTROLLER COMMUNICATIONS is basically a reference document in a textbook style format. It has been divided into functional sections as they relate to the communications functionality of the Yasnac ERC Controller.

Section 3.0 Communication Basics, provides a brief overview of basic data communications functionality and terminology as it applies to the ERC Controller. This section provides a framework for the following sections, which detail the ERC Controller's specific data communication functions and interface.

Section 4.0 ERC Controller Communications, introduces the devices which have been integrated with the ERC Controller. A separate section is provided for each device which details the communications interface, functions, commands and transaction format between the ERC Controller and that device. This section also provides a description of the ERC Controller's communications interface.

Section 5.0 Floppy Disk Controller, describes the functions provided by the Yasnac FC1 Floppy Disk Controller. An overview of the Floppy Disk Controller's communications interface is described within this section.

Section 6.0 Remote Computer Communications, provides a complete description of the interface between the ERC Controller and a remote computer. A micro (PC), mini or mainframe can be used to provide remote computer functionality to the ERC Controller. A list of available commands and the format recognized by the ERC Controller, for functions such as data file transfer and robot operation, is provided. The customer can utilize this information to develop an application program to communicate with the ERC Controller to provide functions such as the transfer of file data, operation of the robot, or inquiry of the robot's status.

Section 7.0 ERC Communications With A Personal Computer (PC), describes the specific communications interface between the ERC Controller and a personal computer.

Appendix A Glossary, provides definitions of computer hardware/software and data communications terminology used within this document.

1.4 TRADEMARK ACKNOWLEDGMENTS

MOTOMAN has made every attempt to supply trademark information about company names and products mentioned within this document. The use of trade names is in no way an endorsement of the mentioned products or companies.

IBM, PC, PC AT, PC XT and BASICA are registered trademarks of International Business Machines Corporation.

MS-DOS is a registered trademark of Microsoft Corporation.

EPSON is a trademark of Epson Corporation.

YASNAC, ERC, FC1, MY-BASIC, and MOTOMAN are registered trademarks of YASKAWA Electric Manufacturing Company, Limited.

2.0 SAFETY

2.1 STANDARD CONVENTIONS

This manual includes information essential to the safety of personnel and equipment. As you read through this manual, be alert to the four signal words: **DANGER**, **WARNING**, **CAUTION**, and **NOTE**. Pay particular attention to the information provided under these headings which are defined below (in descending order of severity).

➪ *DANGER!*

Information appearing under the DANGER caption concerns the protection of personnel from the immediate and imminent hazards which, if not avoided, will result in immediate, serious personal injury or loss of life in addition to equipment damage. Examples are: Exposed high voltage, open fans, or pinch points.

➪ *WARNING!*

Information appearing under the WARNING caption concerns the protection of personnel and equipment from potential hazards which can result in personal injury or loss of life in addition to equipment damage. Examples are: Incorrect body position, insufficient ventilation, etc.

➪ *CAUTION!*

Information appearing under the CAUTION caption concerns the protection of equipment, software, and data from hazards which can result in minor personal injury or equipment damage.

NOTE: *Information appearing in a NOTE caption provides additional information which is helpful in understanding the item being explained.*

2.2 CAUTIONS AND DANGERS

All operators, programmers, plant and tooling engineers, maintenance personnel, supervisors and anyone working near the robot must become familiar with the operation of this equipment! All personnel involved with the operation of the equipment must have a good knowledge of its safe operation, working range and potential dangers of operation. All personnel must read and thoroughly understand the manuals and all DANGERs, WARNINGs, CAUTIONs, and the conditions under which they may occur. Plant safety personnel must conduct a risk analysis of the completed system and working conditions prior to its operation.

➪ CAUTION!

System operation may differ due to product improvements, options, equipment, accessories, and programming technique!

Additional guarding, barriers, interlocked access, pressure mats, and other means of operator and area protection may be required depending on system installation or location and system operation. The safe operation of the robot, positioner, auxiliary equipment, and system is ultimately the user's responsibility. The conditions under which the equipment will be safely operated should be reviewed by the user. The user must be aware of the various national codes, RIA safety recommendations and other local codes that may pertain to the installation and use of industrial equipment. Refer to the ANSI / RIA publication R15.06-1986, "Safety Requirements for Industrial Robots" and other industry publications for additional safety guidelines.

➪ DANGER!

Improper wiring can cause severe personal injury or death and can damage the equipment! Only trained personnel familiar with the robot manuals, electrical design and equipment interconnections should be permitted to modify the system.

➪ DANGER!

High voltage can kill! DO NOT touch electrically live parts. Disconnect and LOCKOUT / TAGOUT all electrical connections to equipment before making electrical connections.

➪ *DANGER!*

Check all safety equipment frequently for proper operation! Repair any non-functioning safety equipment immediately.

➪ *DANGER!*

This equipment has multiple sources of electrical supply! Electrical interconnections are made between the ERC controller, External Servo Box, and other equipment. Disconnect and Lockout all electrical circuits before making modifications / connections.

The ERC Controller allows the customer to modify its internal logic for maximum robot performance. Considerable care must be taken when making these modifications. Double check all modifications under every mode of operation to insure that you have not created hazardous or dangerous situations that may cause personal injury or damage the robot or other parts of the system.

➪ *DANGER!*

All modifications made to the ERC Controller will change the way the robot operates and can cause severe personal injury or death and can damage the robot! This includes ERC parameters; LADDER NODES 1, 2, or 3; and I/O modifications. Recheck and test all changes at slow speed.

➪ *DANGER!*

Any person in the robot cell during teaching must have the robot teach pendant in their hand!

➪ *DANGER!*

Improper operation can damage equipment! Only trained personnel familiar with the operation of this MOTOMAN robot, the operator's manuals, the system equipment, options, and accessories should be permitted to operate this robot system.

➪ *DANGER!*

Do not enter the robot cell while it is in operation. Place the robot in E-STOP mode and check that all motion has stopped before entering the cell.

Read and understand the following messages and the other messages used throughout this manual.

➪ *CAUTION!*

Improper connections can damage the robot! All connections must be made within the standard voltage and current ratings of the robot I/O.

➪ *CAUTION!*

The robot must be placed into E-STOP mode whenever it is NOT in use!

NOTE: *Back up all your programs and jobs onto a floppy disk whenever program changes are made! A backup must always be made before any servicing or changes are made to options, accessories, or equipment to avoid loss of information, programs, or jobs.*

3.0 COMMUNICATION BASICS

3.1 INTRODUCTION

In order for computing devices to communicate there must be an interface, i.e. transfer media (such as a cable), through which information can flow. Additionally, a common set of rules must be used if the devices are to understand each other. Protocol is a term which describes the set of procedures and conventions used to formalize information transfer and error control. The following sections provide a basic description of communication terminology as it applies to the ERC Controller.

3.2 BINARY DATA ENCODING

Computer data streams are simply a sequence of electrical highs and lows (1s and 0s). A simple eight-bit sequence can produce any number between 0 and 255. These 256 values have been assigned to character sets. Thus, every time a computer receives a certain value, it may be interpreted as a unique character.

More than one of these code sets exists, but the most common is ASCII (American Standard Code for Information Interchange). ASCII was developed by the American National Standards Institute (ANSI) and has attained nearly universal acceptance. EBCDIC (Extended Binary Coded Decimal Interchange Code), another code set, has been used widely in IBM applications.

Most people find eight-bit binary values difficult to read. Therefore, we usually look at the numeric representation of the character in hexadecimal (base 16) or decimal (base 10) form. Binary numbers are comprised of 1s and 0s; decimal numbers are comprised of numbers 0 through 9; and hexadecimal numbers are comprised of the numbers 0 through 9, plus the letters A, B, C, D, E and F. (A corresponds to decimal 10, B to 11, C to 12, D to 13, E to 14 and F to 15.) Table 3-1, on the following three pages, provides a cross reference between ASCII character sets and their decimal, hexadecimal, and binary representations.

NOTE: Table 3-1 only provides the first 127 ASCII characters (i.e., those characters which can be depicted with seven (7) bits).

Table 3-1 ASCII Character Set and Cross Reference

DECIMAL	HEXADECIMAL	BINARY	ASCII MEANING
0	0	00000000	NUL
1	1	00000001	SOH
2	2	00000010	STX
3	3	00000011	ETX
4	4	00000100	EOT
5	5	00000101	ENQ
6	6	00000110	ACK
7	7	00000111	BEL
8	8	00001000	BS
9	9	00001001	HT
10	A	00001010	LF
11	B	00001011	VT
12	C	00001100	FF
13	D	00001101	CR
14	E	00001110	SO
15	F	00001111	SI
16	10	00010000	DLE
17	11	00010001	DC1
18	12	00010010	DC2
19	13	00010011	DC3
20	14	00010100	DC4
21	15	00010101	NAK
22	16	00010110	SYN
23	17	00010111	ETB
24	18	00011000	CAN
25	19	00011001	EM
26	1A	00011010	SUB
27	1B	00011011	ESC
28	1C	00011100	FS
29	1D	00011101	GS
30	1E	00011110	RS
31	1F	00011111	US
32	20	00100000	Space
33	21	00100001	!
34	22	00100010	"
35	23	00100011	#
36	24	00100100	$
37	25	00100101	%
38	26	00100110	&
39	27	00100111	'
40	28	00101000	(
41	29	00101001	)
42	2A	00101010	*
43	2B	00101011	+
44	2C	00101100	,
45	2D	00101101	-
46	2E	00101110	.

47	2F	00101111	/

Table 3-1 ASCII Character Set and Cross Reference Continued

DECIMAL	HEXADECIMAL	BINARY	ASCII MEANING
48	30	00110000	0
49	31	00110001	1
50	32	00110010	2
51	33	00110011	3
52	34	00110100	4
53	35	00110101	5
54	36	00110110	6
55	37	00110111	7
56	38	00111000	8
57	39	00111001	9
58	3A	00111010	:
59	3B	00111011	;
60	3C	00111100	<
61	3D	00111101	=
62	3E	00111110	>
63	3F	00111111	?
64	40	01000000	@
65	41	01000001	A
66	42	01000010	B
67	43	01000011	C
68	44	01000100	D
69	45	01000101	E
70	46	01000110	F
71	47	01000111	G
72	48	01001000	H
73	49	01001001	I
74	4A	01001010	J
75	4B	01001011	K
76	4C	01001100	L
77	4D	01001101	M
78	4E	01001110	N
79	4F	01001111	O
80	50	01010000	P
81	51	01010001	Q
82	52	01010010	R
83	53	01010011	S
84	54	01010100	T
85	55	01010101	U
86	56	01010110	V
87	57	01010111	W
88	58	01011000	X
89	59	01011001	Y
90	5A	01011010	Z

Table 3-1 ASCII Character Set and Cross Reference Continued

DECIMAL	HEXADECIMAL	BINARY	ASCII MEANING
91	5B	01011011	[
92	5C	01011100	\
93	5D	01011101	]
94	5E	01011110	^
95	5F	01011111	_
96	60	01100000	`
97	61	01100001	a
98	62	01100010	b
99	63	01100011	c
100	64	01100100	d
101	65	01100101	e
102	66	01100110	f
103	67	01100111	g
104	68	01101000	h
105	69	01101001	i
106	6A	01101010	j
107	6B	01101011	k
108	6C	01101100	l
109	6D	01101101	m
110	6E	01101110	n
111	6F	01101111	o
112	70	01110000	p
113	71	01110001	q
114	72	01110010	r
115	73	01110011	s
116	74	01110100	t
117	75	01110101	u
118	76	01110110	v
119	77	01110111	w
120	78	01111000	x
121	79	01111001	y
122	7A	01111010	z
123	7B	01111011	{
124	7C	01111100	|
125	7D	01111101	}
126	7E	01111110	~
127	7F	01111111	DEL

3.3 INTERFACE (SERIAL VS PARALLEL)

The connection media between computing devices provides the interface (avenue for information flow) and thus, determines the amount of simultaneous data which may be transferred. Two interface methods; serial and parallel, can be used to transfer information between computing devices.

3.3.1 Serial

In serial transmission, information is transferred, one data bit at a time over a single transmission line. The flow of this data can follow one of three transmission modes: simplex, half duplex or full duplex. Simplex allows data flow in one direction only: half duplex allows data flow in both directions (but not simultaneously), while full duplex allows simultaneous two-way transmission. Examples of simplex, half-duplex and full-duplex are provided in Figure 3.1, below.

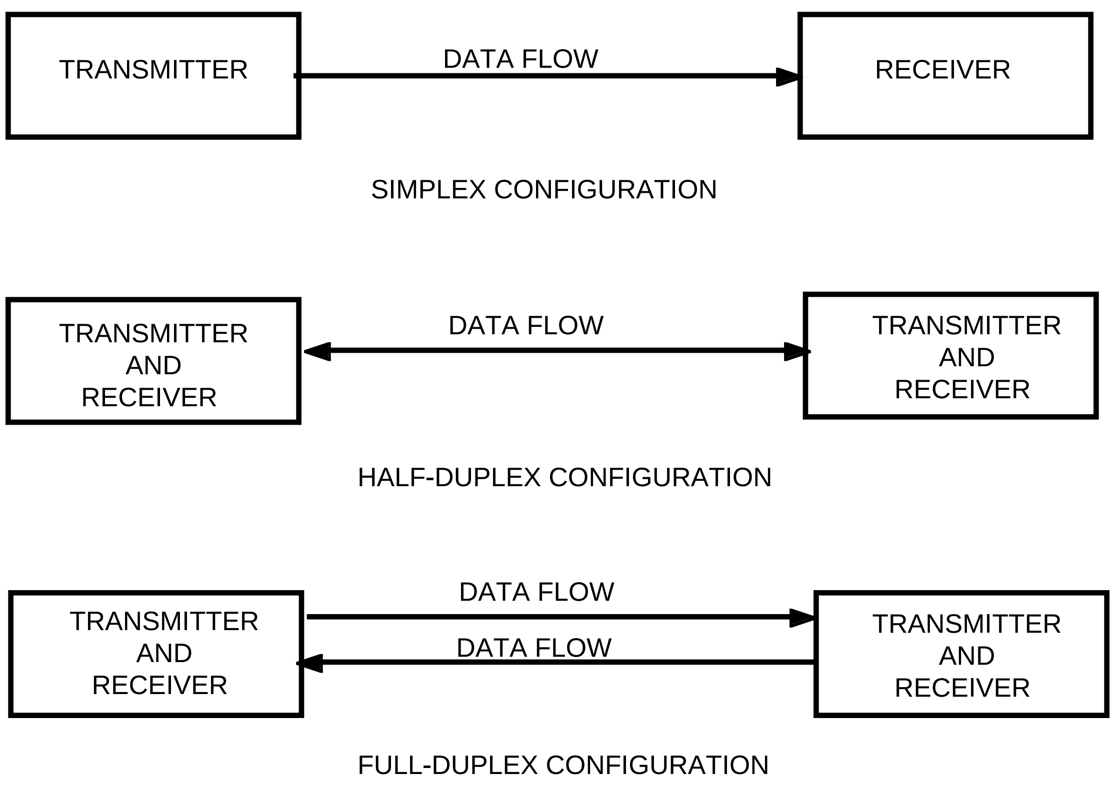


Figure 3.1 Data Transmission Modes

3.3.1.1 RS-232 and RS-449 Interface

RS-232 is an Electronic Industry Association (EIA) standard applicable to the 25-pin interconnection of Data Terminal Equipment (DTE) and Data Communication Equipment (DCE) employing a serial binary connection. Figure 3.2 illustrates the RS-232 interface.

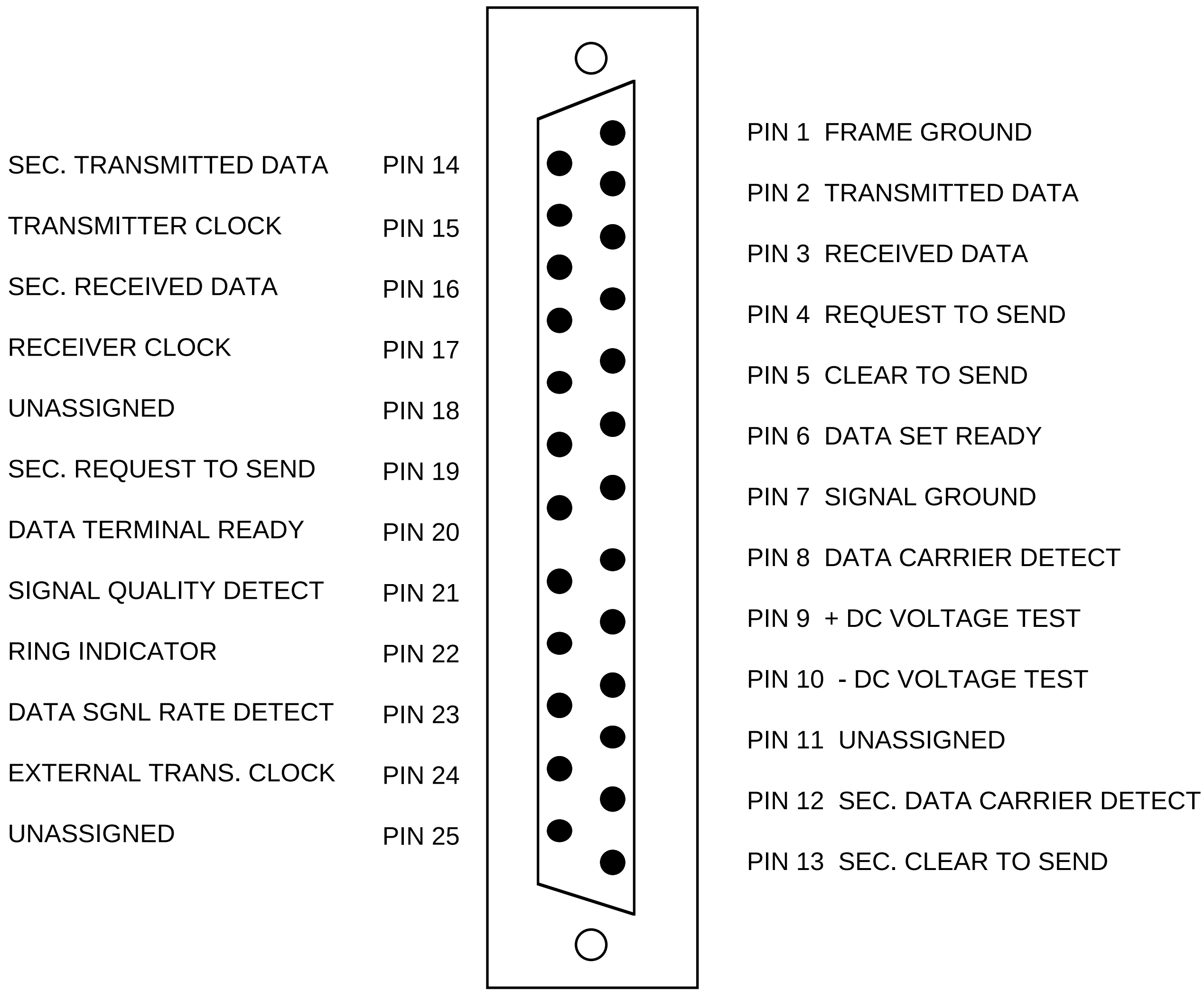


Figure 3.2 RS-232 Interface

3.3.1.2 Pinout Description For RS-232C Interface

Table 3-2 Pinout Description for RS-232C Interface

PIN NUMBER	RS-232 DESCRIPTION
1	FG, Frame Ground
2	TD, Transmitted Data
3	RD, Received Data
4	RTS, Request To Send
5	CTS, Clear To Send
6	DSR, Data Set Ready
7	SG, Signal Ground
8	DCD, Data Carrier Detect (Received Line Signal Detector)
9	Positive DC Test Voltage
10	Negative DC Test Voltage
11	Unassigned
12	(S)DCD, Secondary Data Carrier Detect
13	(S)CTS, Secondary Clear To Send
14	(S)TD, Secondary Transmitted Data
15	TC, Transmitter Clock
16	(S)RD, Secondary Received Data
17	RC, Receiver Clock
18	Unassigned
19	(S)RTS, Secondary Request To Send
20	DTR, Data Terminal Ready
21	SQ, Signal Quality Detect
22	RI, Ring Indicator
23	Data Signal Rate Select
24	External Transmitter Clock
25	Unassigned

RS-449 is an EIA standard, applicable to the 37-pin and 9-pin interconnection of DTE and DCE devices that use the serial binary data exchange. Since the RS-449 interface does not relate to the ERC control, no further description is provided in this manual.

3.3.2 Parallel

Parallel transmission involves the transfer of data bits over multiple transmission lines simultaneously. A parallel interface is most commonly used in computer-to-printer applications. Further discussion of a parallel interface will not be provided in this document.

3.4 DATA TRANSMISSION

Once the interface has been established, a method of controlled data transmission must be employed. Data transfer between devices may be asynchronous or synchronous. Asynchronous data transfer occurs randomly while synchronous data transfer occurs at timed intervals.

3.4.1 Asynchronous

Within the asynchronous data stream, each character of data is transported as a binary bit frame. Each frame begins with a start bit which is a low voltage signal on the data line. Upon detection of the start bit, the receiving device can begin looking for 0s and 1s which constitute the data bits. The following five to eight data bits (the number depends on the code used) comprise the binary character. For error detection an optional parity bit can mark whether the total number of 0s or 1s was odd or even. An explanation of the parity types and associated meanings is provided in Figure 3.4. A stop bit signals the end of the character. Stop bits range in length from one to two bits. Generally, the slower the transmission, the more stop bits required for end-of-frame recognition. Figure 3.3 depicts an asynchronous data bit frame.

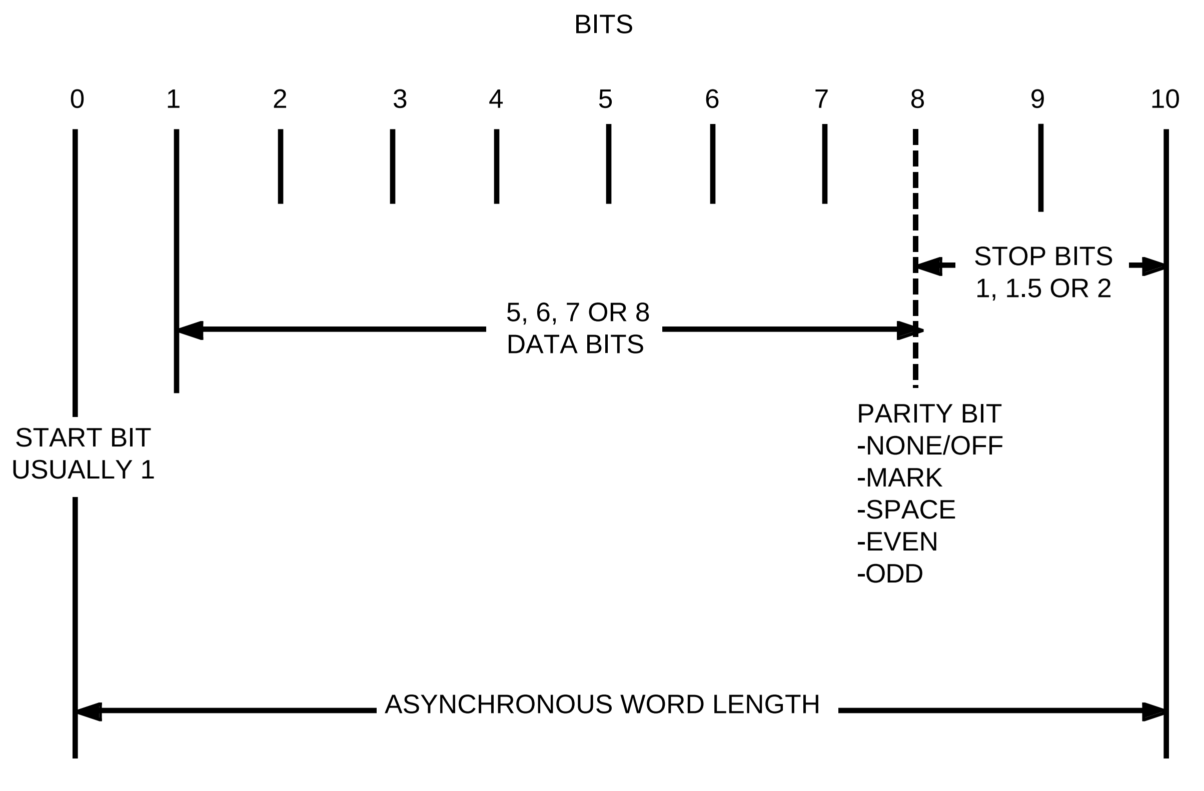


Figure 3.3 Asynchronous Bit Frame

PARITY TYPE	DESCRIPTION
ODD	EIGHTH DATA BIT IS LOGICAL ZERO IF THE TOTAL NUMBEF OF LOGICAL 1S IN THE FIRST SEVEN DATA BITS IS ODD
EVEN	EIGHTH DATA BIT IS LOGICAL ZERO IF THE TOTAL NUMBEF OF LOGICAL 1S IN THE FIRST SEVEN DATA BITS IS EVEN
MARK	EIGHTH DATA BIT IS ALWAYS LOGICAL 1 (HIGH/MARK)
SPACE	EIGHTH DATA BIT IS ALWAYS LOGICAL 0 (LOW/SPACE)
NONE/OFF	EIGHTH DATA BIT IS IGNORED

Figure 3.4 Parity Bit Description

The previous paragraph describes the data frame which is used to constitute a character of information. In order for flow to occur without loss of data, each end of the data link must conform to certain operating parameters.

The first parameter is the rate of transmission. Serial data transmission is measured in bits per second (bps). Typical asynchronous serial transmission rates are 110, 300, 600, 1200, 2400, 4800, 9600 and 19200 bps. In order for two devices to interact, both must be operating at the same transmission rate, or an intermediate memory device, called a buffer, that accommodates the differences in speed, must be used

In addition to transmit and receive buffers, most serial asynchronous links use a flow control system to handle data transmission. A common flow control is X-ON/X-OFF. When a receive buffer reaches its memory capacity, the receiving device sends an ASCII DC3 (X-OFF) signal to the transmitting device telling it to stop sending data. When the receive buffer has sufficiently unloaded enough data, it transmits an ASCII DC1 (X-ON) signal back to the transmitting device, telling it to continue transmission.

Other flow controls commonly used in asynchronous communications are lead control and the ENQ/ACK protocol. A lead control protocol controls data flow by alternately raising and lowering the voltage on a lead of the RS-232 interface. DTR (pin 20) and CTS (pin 5) are often used. When the voltage on a pin is high, data flow is enabled; when the voltage drops low, data flow is stopped.

ENQ/ACK is a block-oriented protocol, i.e. a protocol that sends a fixed amount of characters every time it transmits. Typically, the transmitting device sends an

ENQ character and waits for an ACK character from the receiving station before beginning transmission. Once the ACK character is received, the entire block is transmitted.

3.4.2 Synchronous

In synchronous data transmission, special characters synchronize the transmitting and receiving elements of the link. This provides a benefit in that transmissions occur without the overhead of start and stop bits, as in asynchronous communication.

Synchronous protocols fall into two major categories: 1) character (or byte) oriented which specify a definite character length and 2) bit oriented which do not specify character boundaries.

BSC, which is short for Binary Synchronous Communication (or Bisynchronous) is one of the most common character-oriented protocols. BSC uses a set of special characters to define each structure of the data transmission frame.

SDLC, which is short for Synchronous Data Link Control is the typical bit-oriented protocol. SDLC uses a variety of bit patterns to flag the beginning and end of a frame. Other bit patterns are used for the address, control and packet header fields which route the frame through the network to its destination.

The ERC Controller utilizes asynchronous data transmission. Therefore, in order to distinguish the difference between the two types of data transmission methods, only a brief description of synchronous data transmission has been provided.

4.0 ERC CONTROLLER COMMUNICATIONS

4.1 ERC COMMUNICATION FUNCTIONS

The ERC Controller is capable of communicating with external devices to transmit/receive commands or transfer data. These devices can include floppy disk controllers, micro (personal), mini and mainframe computers, or other equipment capable of transmitting and receiving data via the defined interface and protocol.

4.2 ERC CONTROLLER COMMUNICATIONS INTERFACE

The Operator's Subpanel on the ERC Controller features a DB-25 female connector (labeled RS-232C), with pin assignments following the RS-232C standard, for communication with external devices.

NOTE: *Not all twenty-five (25) pins are connected within the ERC Controller.*

Figure 4.1 on the following page illustrates the pins which are connected and utilized by the ERC Controller.

➪ *CAUTION!*

The fitting on the ERC Operator's Subpanel DB-25 female connector requires metric screws. Use of non-metric screws may damage the threads of the DB-25 connector.

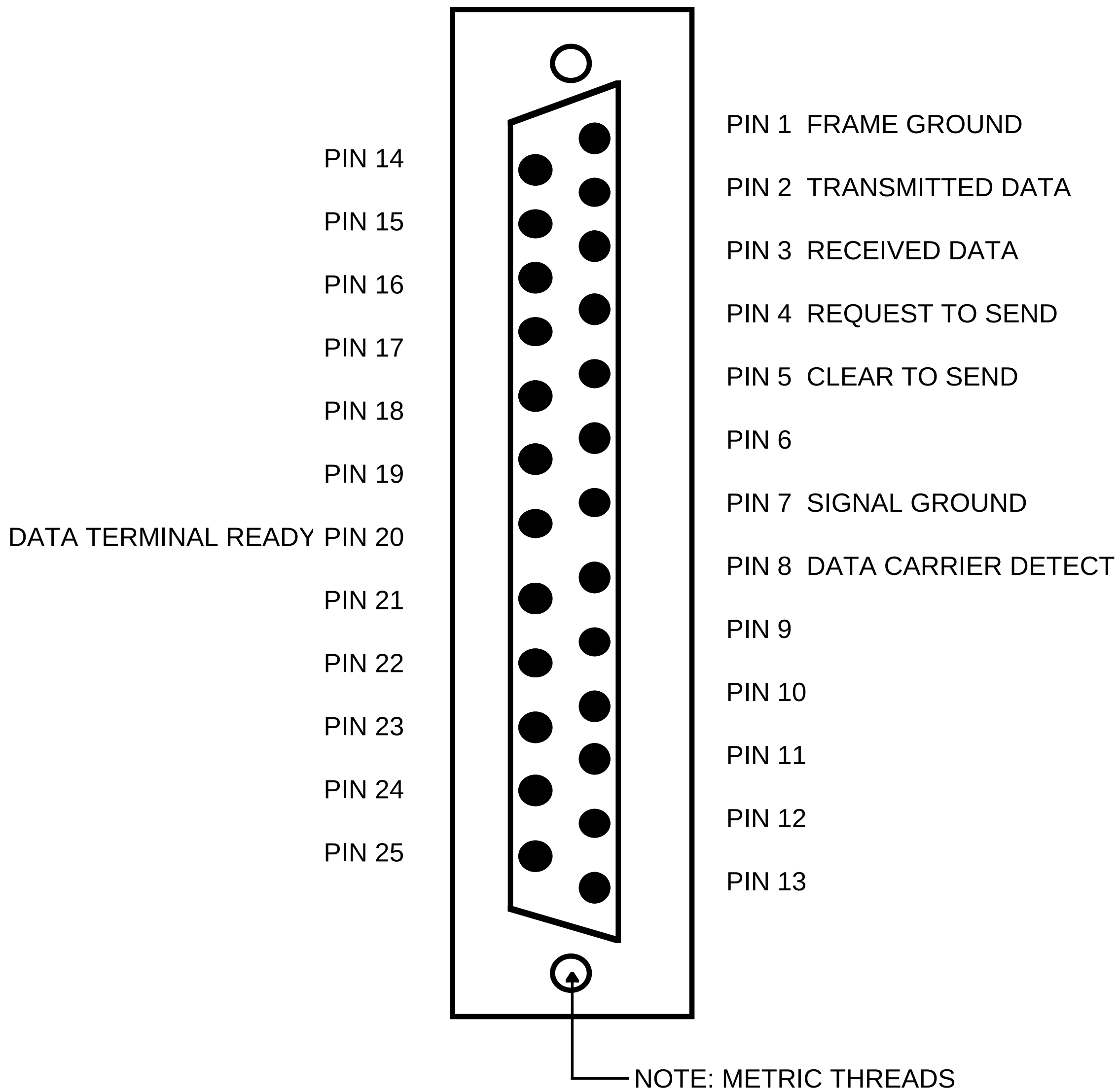


Figure 4.1 ERC Controller's DB-25 Female Connector

4.3 ERC CONTROLLER COMMUNICATIONS PROTOCOL

In order to provide for data flow both to and from the ERC, Half-Duplex transmission mode is utilized. The ERC Controller recognizes the ASCII character set and the ENQ/ACK block-oriented protocol set. Although asynchronous data transmission is the communication method, the ERC Controller recognizes commands which are sent in a BSC-like character-oriented format. The communication parameters: bps (bits per second), parity, and number of data and stop bits on the ERC Controller are selectable and changeable. Normally, once these values have been established and set, they are not altered unless the device to which the ERC is communicating with changes. Specific parameter values for each device are provided within the section of this document which describes that device. The command transaction formats between the ERC Controller and each external device are also provided within the device-specific section.

5.0 FLOPPY DISK CONTROLLER

5.1 INTRODUCTION

The Yasnac Model FC1 Floppy Disk Controller is a rugged, portable device which has been designed to operate with the ERC Controller. The purpose of this device is to provide a user-friendly, reliable method of saving/loading jobs and files residing on floppy disks to/from the ERC Controller.

The purpose of this manual is to provide an overview description of the communications between the ERC Controller and the FC1 Floppy Disk Controller. In addition to the information supplied in this manual, four instruction manuals for the Yasnac Model FC1 Floppy Disk Controller are available. Consult the appropriate manual for protocol information, operation and instructions regarding the FC1 Floppy Disk Controller. The title and MOTOMAN part number for each manual are provided in Table 5-1, below.

5.1.1 Yasnac Model FC1 Floppy Disk Controller Manuals

Table 5-1 Yasnac Model FC1 Floppy Disk Controller Manuals

TITLE	**PART NUMBER**
Instructions For Floppy Disk Controller With ERC	479236-12
Instructions For Floppy Disk Controller With RX	479236-6
FC1 Floppy Disk Printer Function	479236-8

The following sections describe the communications between the ERC Controller and the Yasnac Model FC1 Floppy Disk Controller.

5.2 REQUIREMENTS

The following items are required in order to load/save files from the ERC Controller to floppy disks using the FC1:

- Yasnac Model FC1 Floppy Disk Controller
- Communications Cable
- 3.5 Inch Floppy Disks

5.2.1 YASNAC Model FC1 Floppy Disk Controller

The following specifications correspond to the FC1 Floppy Disk Controller.

Manufacturer:	Yasnac
Model:	FC1
CPU:	64180
Floppy Disk Drive (FDD):	3.5 inch 640/720 K
LCD:	Sixteen (16) Characters wide, Two (2) characters high
Serial Ports:	Two (2) designated CON1 (connector 1) and CON2 (connector 2)
Memory:	64K ROM; 16K RAM; EEPROM for FC1
Keyboard:	Membrane type; Ten (10) digits, Eighteen (18) functions
Power:	85-125 VAC, 50/60 Hz, 13 Watt
Size:	8.7"x5.6"x9.5"
Weight:	9.4 Lbs.
Operating conditions:	
Temperature:	5 °C to 40 °C
Humidity:	20%-80% Relative Humidity
Vibration:	1 G
Shock:	5 G
Storage conditions:	
Temperature:	-20 °C to 60 °C
Humidity:	No condensation
Vibration:	2 G
Shock:	50 G
Atmosphere:	No corrosive, explosive gases, or oil mist
Printer Interface:	CON2 (Use Epson-compatible serial printer)

Figures 5.1 and 5.2 provide both a front and rear view of the Yasnac Model FC1 Floppy Disk Controller.

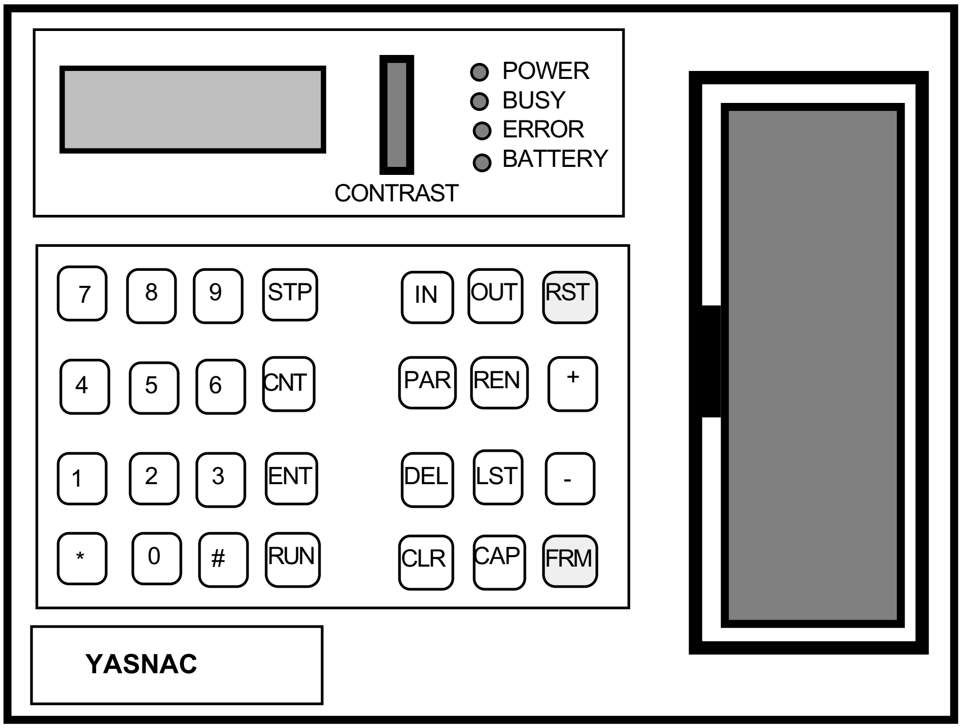


Figure 5.1 Yasnac Model FC1 Floppy Disk Controller - Front View

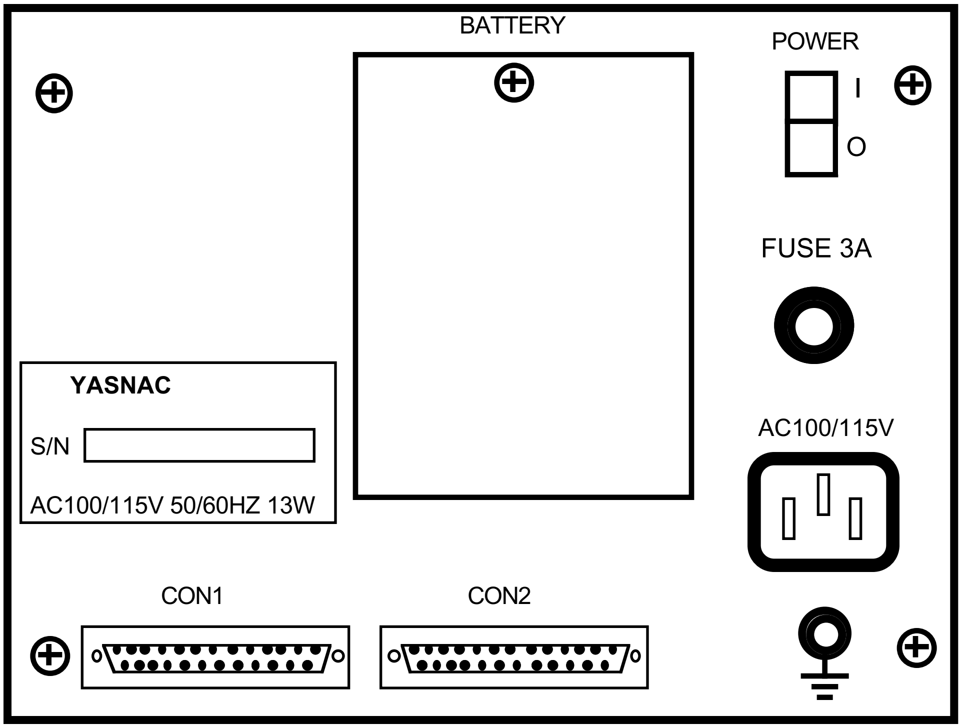


Figure 5.2 Yasnac Model FC1 Floppy Disk Controller - Rear View

5.2.2 Communications Cable

A communications cable connects the RS-232C port on the ERC Controller to the CON1 port on the FC1. CON1 is a DB-25 female connector, as is the connector on the ERC Controller. The cable connecting the two devices follows the RS-232C standard for pin assignments. However, not all twenty-five (25) pins are used. Therefore, a 25-wire cable is not required. Figure 5.3 illustrates the required cable connections.

CAUTION!

The ends of the cable are different. Cables purchased through Motoman are labeled FDD (for connection to the FC1) and with an asterisk (*) (for connection to the ERC Controller). If a new cable is made, label which end is connected to the ERC Controller and which is connected to the FC1. Failure to connect the cable correctly may result in loss of data or improper operation of the FC1.

➪ *CAUTION!*

The RS-232C standard limits the length of the cable to 50 feet. A longer cable may be used but should be thoroughly tested in the working environment to ensure signal quality.

➪ *CAUTION!*

To ensure signal quality, always use a shielded cable.

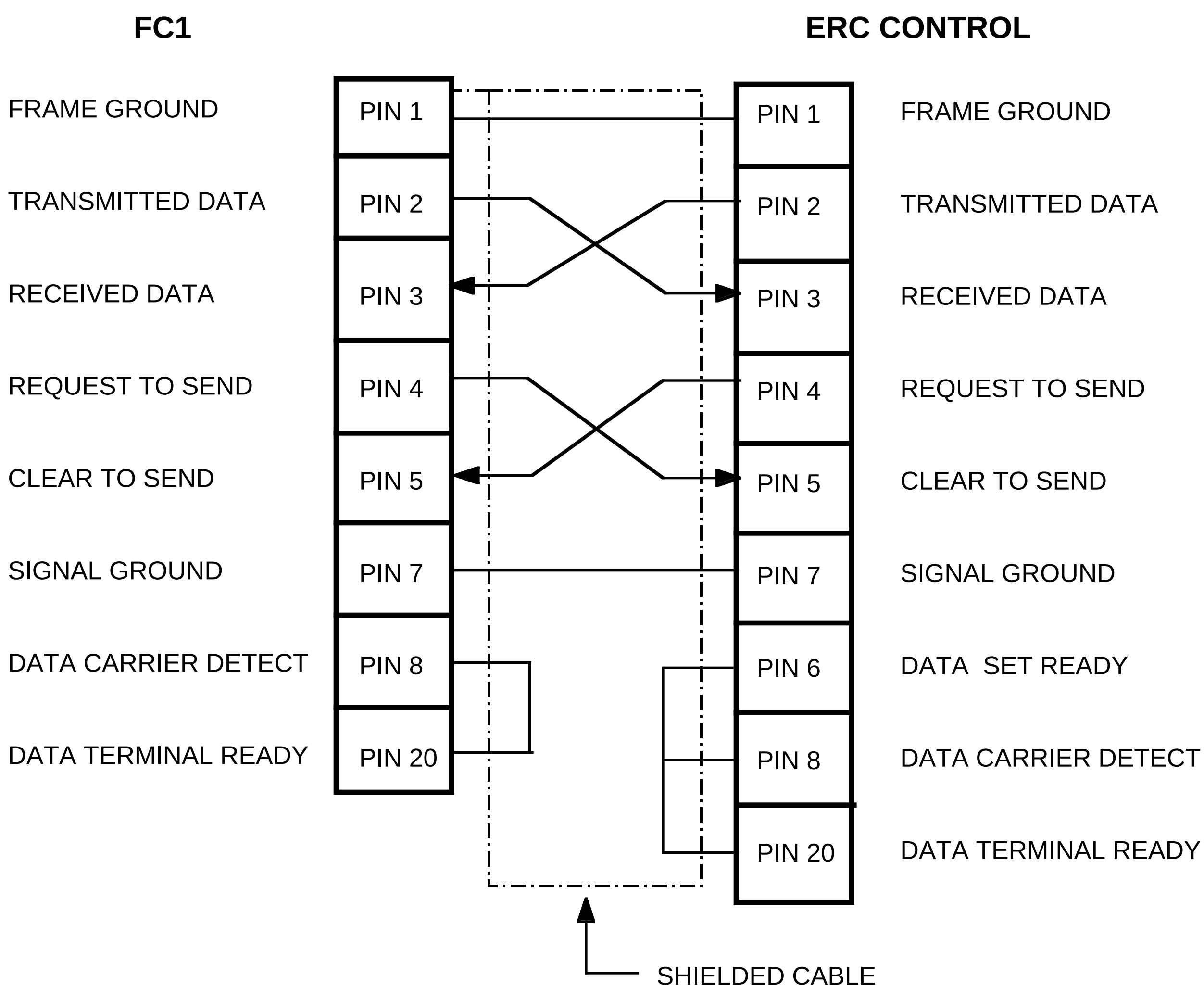


Figure 5.3 ERC to FC1 Communications Cable

5.2.3 Floppy Disks

The FC1 is a 3.5 inch floppy disk drive that uses 3.5 inch double-sided, double-density (denoted DSDD) floppy disks. A DSDD floppy disk will format to approximately 720K of usable space. Double-sided, high capacity or double-sided, high-density disks (denoted DSHC and DSHD) floppy disks **cannot** be read by the FC1. Floppy disks are available from many manufacturers.

➪ *CAUTION!*

Do not use high capacity (1.44 M) floppy disks. These disks cannot be read by the FC1.

5.3 COMMUNICATION SPECIFICATIONS

The interface between the FC1 Floppy Disk Controller and the ERC Controller conforms to the RS-232C standard for pin assignments (refer to Section 3.2.2 Communications Cable). The following specifications are employed for data communications between the FC1 Floppy Disk Controller and the ERC Controller.

Interface:	RS-232C compatible
Transmission Mode:	Serial Asynchronous Half-Duplex
Transmission Code:	ASCII
Data Bits:	Eight (8)
Parity:	Even
Stop Bits:	One (1)
Transmission Speed:	4800 bps
Flow Control:	ENQ/ACK

To set or check the communication parameters on the FC1, follow the directions provided in the Instructions For Floppy Disk Controller With ERC manual (Motoman Part Number 479236-12.) To set or display the communication parameters on the ERC Controller, follow the directions provided in the K Series Robot Operator's Manual.

5.4 COMMAND TRANSACTION FORMAT

The command transaction format is described in FC1 Communications Protocol (Motoman Part Number 479236-23).

6.0 REMOTE COMPUTER COMMUNICATIONS

6.1 INTRODUCTION

This section contains information on communication functions between the Yasnac ERC Controller and a remote computer. In this context, "remote" refers to any computer which communicates with the ERC Controller through the DB-25 serial port connection (labeled RS-232C) on the ERC Controller. Remote computers may include micro (personal), mini and mainframe computers.

The connection of a remote computer provides a variety of functions. The ERC Controller can benefit from increased data storage capacity, through the use of the remote computer's disk drives, for the storage of Job, System and Condition data. Job data can be created off-line on the remote computer and downloaded to the ERC Controller. The remote computer can be used to control the operation of the robot. Information on the robot's status and position can be sent to the remote computer for inspection. Job and variable data may be automatically transferred to and from the robot to meet working conditions.

6.2 REQUIREMENTS

The following items are required for ERC communication with a remote computer:

- Micro (personal), mini or mainframe computer with serial communications port.
- Communications cable.
- Yasnac ERC Controller Data Communications Option (Part Number 479323).
- A programmer with data communications experience.

6.2.1 Micro, Mini, or Mainframe Computer

The type, size or brand of computer used to communicate with the ERC Controller is irrelevant. However, the computer must be able to interface to the ERC Controller through a serial port connection described in Section 6.2.2. Additionally, it must be able to send and receive data, and run application software which is compatible with the protocols described within the remainder of Section 6.

6.2.2 Communications Cable

A DB-25 female connector (labeled RS-232C) on the ERC Controller Operator's Subpanel provides an interface with external devices (e.g. remote computers, FC1 Floppy Disk Controller, etc.). The DB-25 connector follows the RS-232C standard for pin assignments. In order to transfer serial data, a communications cable is required to connect the remote computer's serial port to this DB-25 connector. Figure 6.1, on the following page, illustrates the required cable connections.

NOTE: Not all twenty-five (25) pins on the ERC's DB-25 connector are used. Therefore, the connecting cable does not require all twenty-five (25) wires.

➪ ***CAUTION!***

The ends of the cable are usually different. If a new cable is made, label which end is connected to the ERC Controller and which is connected to the remote computer. Failure to connect the cable correctly may result in loss of data or improper transmission.

➪ ***CAUTION!***

The RS-232C standard limits the length of the cable to 50 feet. A longer cable may be used, but should be thoroughly tested in the working environment to ensure signal quality.

➪ ***CAUTION!***

To ensure signal quality, always use a shielded cable.

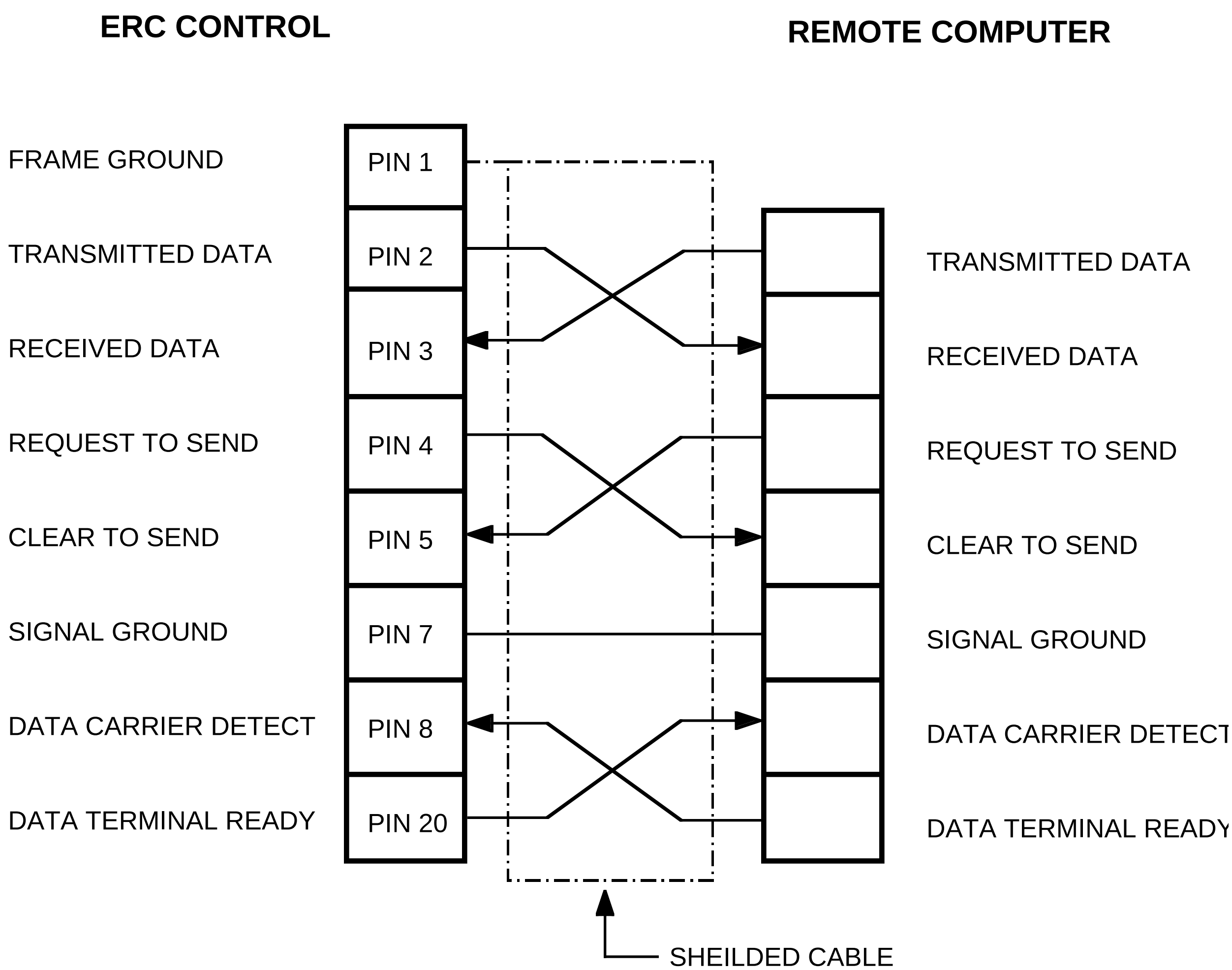


Figure 6.1 ERC to Remote Computer Communications Cable

Figure 6.1 depicts the required connections between the ERC Controller and a remote computer. Because the connector pin assignments on the remote computer will vary depending upon the device, pin numbers for the remote computer are not specified. Instead, only the functionality of the required connections is provided. For specifics regarding the remote computer's connection pin assignments, refer to information supplied with the remote computer.

The following descriptions are provided for the ERC Controller's pin assignments.

Pin 2 (Transmitted Data) is used to transmit data from the ERC Controller. This pin should be connected to the receiving pin on the remote computer.

Pin 3 (Received Data) is used by the ERC Controller to receive data. This pin should be connected to the transmission pin on the remote computer.

Pin 4 (Request To Send) is used by the ERC Controller to indicate that it is ready to receive additional data. This pin should be connected to the pin(s) on the remote computer which control the transmission of data. The ERC will drop the signal on this pin when it is in a state in which it can not receive any more data (e.g. receive buffer full).

Pin 5 (Clear To Send) is used to signal the ERC Controller that the remote computer is ready to receive data from the ERC Controller.

Pin 7 (Signal Ground) is used to provide a common zero reference voltage between the ERC Controller and the remote computer.

Pin 8 (Data Carrier Detect) on the ERC Controller is provided to maintain consistency of connections. However, the ERC Controller does not utilize the input signal received on this pin.

Pin 20 (Data Terminal Ready) on the ERC Controller is output whenever the ERC Controller is capable of transmitting or receiving data.

The required cable will have a DB-25 male connector end for connection to the ERC. The connection end for the remote computer will vary depending on the device. The majority of serial ports have a DB-25 male connector which requires a cable with a DB-25 female connector. In the case of the IBM PC AT, the standard serial port is a DB-9 male connector. This will require a DB-9 female connector on the end of the cable attached to the remote computer. Alternatively, a DB-9 to DB-25 adapter may be available through most computer stores.

6.2.3 YASNAC Data Communications Option

For the remote computer to communicate with the ERC, the Data Communications Option (Part Number 479323-1) must be purchased for the ERC Controller. This option provides the capability to operate the robot from the remote computer, check the robot's status and position, as well as transfer job, condition and system data.

Additional functionality can be provided to the basic Data Communications Option with the inclusion of the DCI (Data Communications by Instruction) Function. This function provides the capability to automatically transfer jobs and variable data between the ERC Controller and a remote computer through the use of INFORM programming instructions. This option is known as Data Communications Option with DCI (Part Number 479323-3).

The Data Communications Option consists of communications software, contained on ROM (Read Only Memory) chips mounted on the JANCD-MM13 board in the ERC Controller. This option is factory installed or requires a service call to install in the field.

6.2.4 Programmer With Data Communications Experience

A person with knowledge of the remote computer and data communications will certainly decrease the time to set up and test the interface between the remote computer and the ERC Controller.

Although the communications interface, protocols and command transaction formats are provided, the customer must utilize this information to develop an application program on the remote computer to communicate with the ERC Controller. The application program would typically be developed in a high-level language such as PASCAL, BASIC, FORTRAN, or C.

A reduction in the resources and time required to develop an application program interface can be achieved through the implementation of an existing package. MY-BASIC, an application program, is available from MOTOMAN for operation on IBM PC XT or AT computers.

➪ *DANGER!*

Improper programming can cause serious personal injury or death and damage to the robot. Check all new programs and/or any changes to programs on the robot in slow speed.

➪ *WARNING!*

Improper programming can cause unexpected robot motion. Test all programs thoroughly. Only programmers with serial communication and remote machine control experience should be allowed to use the Data Communication Function.

6.3 COMMUNICATIONS

6.3.1 Communication Specifications

The interface between the ERC Controller and the remote computer conforms to the RS-232C standard for pin assignments (refer to Section 6.2.2 Communications Cable). The following specifications are employed for data communications between the ERC Controller and a remote computer.

Interface:	RS-232C Compatible
Transmission Mode:	Serial Asynchronous Half-Duplex
Transmission Code:	ASCII
Data Bits:	Eight (8)
Parity:	None
Stop Bits:	One (1)
Transmission Speed:	9600 bps
Flow Control:	ENQ/ACK

6.3.2 Communication Parameters

The communications parameters on the ERC Controller are selectable and changeable. These parameters are accessible through the ERC operator's panel OP2 key. The parameters available for communications control are described in Table 6-1 below.

Table 6-1 ERC Controller Communications Parameters

PARAMETER NUMBER	DESCRIPTION
RS00	Number of data bits in the ASCII data field.
RS01	Number of stop bits.
RS02	Parity check.
RS03	Transmission rate (bits per second).
RS06	Watchdog Timer A
RS07	Watchdog Timer B
RS08	Retry 1, Sequence retry counter
RS09	Retry 2, Text retry counter

Table 6-1 ERC Controller Communications Parameters Continued

RS00: The number of data bits can be set to either seven (7) or eight (8). The default value is eight (8).

RS01: The number of stop bits can be set to one (1), one-and-a-half (1.5) or two (2) by the following respective values: 0, 1 or 2. The default value is 0, which is equivalent to one (1) stop bit.

RS02: The parity check can be set to either none, odd or even by the following respective values: 0, 1 or 2. The default value is 2, which is equivalent to even parity.

RS03: The transmission rate can be set to 300, 600, 1200, 2400, 4800 or 9600. The rate is calculated by multiplying the parameter value (RS03) by 150. The following values are valid for the transmission rate: 2, 4, 8, 16, 32, and 64. The default value is 64, which is equivalent to a transmission rate of 9600.

RS06: Watchdog Timer A protects the ERC from an invalid or no response from the remote computer. The timer can be set from 0 to 10 seconds in .1 second increments. The timer value is calculated by multiplying the parameter value

(RS06) by .1. The integer numbers from 0 to 100 are valid. The default value is 30, which is equivalent to 3.0 seconds.

RS07: Watchdog Timer B protects the ERC from not receiving an end of text character from the remote computer. The timer can be set from 0 to 25.5 seconds in 0.1 second increments. The timer value is calculated by multiplying the parameter value (RS07) by 0.1. The integer numbers from 0 to 255 are valid. The default value is 200, which is equivalent to 20.0 seconds.

RS08: Retry 1 specifies a number of times the ERC will retry sending a sequence of data after an invalid response or no response. The retry counter may be set from 0 to 30. The default value is 10, which is equivalent to 10 retries.

RS09: Retry 2 specifies a number of times the ERC will retry sending a sequence of data after receiving a NAK or abnormal block check signal. The retry counter may be set from 0 to 10. The default value is 3, which is equivalent to 3 retries.

6.3.3 Communication Controls

The first thirty-two (32) ASCII characters are dedicated as control characters. These characters act as signals to control specific operations of printing, display and communication devices. Control characters are broken down into three groups:

1. Communication Control
2. Forms Effectors
3. Information Separators.

The communication control group is utilized by the ERC for transmission control between the ERC and a remote computer. The communication control characters perform three basic functions:

1. They tell the receiving device what type of data to expect.
2. They indicate a transition in the type of data being transmitted.
3. They are used to verify proper transmission and receipt.

The communication control characters which are utilized by the ERC Controller are described in Table 6-2.

Table 6-2 Communication Control Characters

CONTROL CHARACTER	HEX CODE	DESCRIPTION
SOH	01	Start Of Heading denotes the start of the message heading data block.
STX	02	Start Of Text denotes end of heading and beginning of information data.
ETX	03	End Of Text signals receiving station that all information data has been transmitted.
EOT	04	End Of Transmission indicates the end of transmission of all data associated with a message.
ENQ	05	Enquiry requests a response from the receiving station.
DLE	10	Data Link Escape is used to modify the meaning of a limited number of subsequent characters.
NAK	15	Negative Acknowledge indicates improper communication.
ETB	17	End of Transmission Block indicates the end of a particular block of transmitted data. ETB is used in place of ETX when data are transmitted in two or more blocks.

In addition to the above communication control characters, the ERC utilizes five (5) two-character control sequences. These additional controls are described in Table 6-3.

Table 6-3 Two-Character Communication Control Characters

CONTROL CHARACTER	HEX CODE	DESCRIPTION
ACK0	10,30	Even acknowledgment.
ACK1	10,31	Odd acknowledgment.
WACK	10,6B	Wait acknowledgment.
RVI	10,7C	Reverse interrupt.
TTD	02,05	Temporary transmission delay.

ACK0/ACK1: The acknowledgments alternate between ACK0 and ACK1. These characters are both sent and received by ERC and the remote computer.

WACK: A receiving device signals this character to the sending device to request a hold on the transmission of data. This occurs when the receiving device is temporarily busy or unable to receive data. If the ERC is the sending device it responds immediately to the WACKwith an ENQ. It is necessary to have a time delay (generally two (2)seconds) on the remote computer. If the ERC receives a WACK when it is not transmitting data to the remote computer, the ERC will respond with an EOT instead of an ENQ. The retry counter ignores the WACK.

RVI: A receiving device signals this character to cause the sending device to stop sending data and surrender the right to transmit. If the ERC is the sending device it responds immediately to the RVI with an EOT, finishes the data transmission, and flags an error status.

TTD: A sending device signals this character to the receiving device to request a hold on the reception of data. This occurs when the sending device is temporarily busy or unable to transmit data. If the ERC is the receiving device it responds immediately to the TTD with an NAK. It is necessary to have a time delay (generally two (2) seconds) on the remote computer. The retry counter ignores the TTD.

6.4 COMMAND TRANSACTION FORMAT

The command transaction format recognized by the ERC Controller is similar to the BSC protocol. Figure 6.2, below, provides the format of a command transaction.

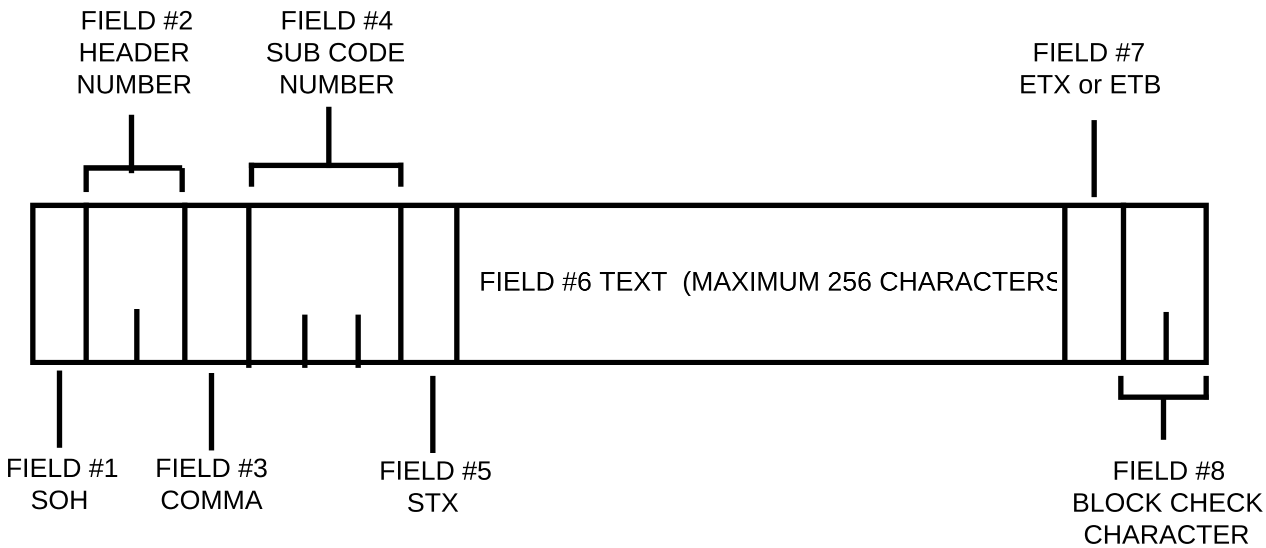


Figure 6.2 ERC Controller Command Transaction Format

Table 6-4 provides a description of the fields within the command transaction.

Table 6-4 Command Transaction Field Contents

FIELD NUMBER	**DESCRIPTION**
1	SOH is an ASCII control character indicating the start of the header in the command transaction.
2	The header number indicates the general type of transaction (e.g. command from host, data, response from host). Field length is two (2) characters. Refer to Section 6.11 for the header code list.
3	A comma is used to separate the header and sub code.
4	The sub code number indicates the specific type of transaction (e.g. weave data, job data, etc.). Field length is three (3) characters. Refer to Section 6.11 for the sub code list.

Table 6-4 Command Transaction Field Contents Continued

FIELD NUMBER	DESCRIPTION
5	STX is an ASCII control character indicating the start of the information data. Field length is variable with a maximum of 256 characters.
6	Text characters constituting the data being transmitted. The maximum number of characters which may betransmitted by a single transaction is 256.
7	ETX or ETB are ASCII control characters indicating the end of the information data.
8	The BCC (block check character) is a method of checking the transaction block for errors. Field length is two (2) characters.

6.5 ERROR DETECTION

The BCC (block check character) provides a method of checking the transaction block for errors. The BCC is a check value, provided by the transmitting station, which allows the receiving station to compare its own calculated check value to the BCC sent by the transmitting station.

The BCC is calculated by using the summation of all the character data transmitted after the SOH or STX control characters. The summation ends when an ETB or ETX control character appears. The summation of the character data is used to create a two (2) character BCC, which is transmitted as the last two characters of the transaction block. The receiving station should keep track of its received characters and calculate its own BCC. To validate the data, the receiving station's calculated BCC is compared to the transmitted BCC. If the two BCCs match, the transmission is error free. The receiving station can then respond with the appropriate ACK0 or ACK1 control characters to indicate a valid reception.

The BCC calculation starts when either an SOH or STX control character appears. These characters are not included in the summation.

NOTE: When an STX appears within a command transaction which starts with an SOH, the STX is included in the BCC calculation.

The BCC calculation ends when an ETB or ETX control character appears. These characters are included in the BCC calculation.

Figure 6.3 provides an example of a BCC calculation for one transaction in the sequence used to save the job VACON from the ERC Controller to a host computer.

BCC Example Calculation

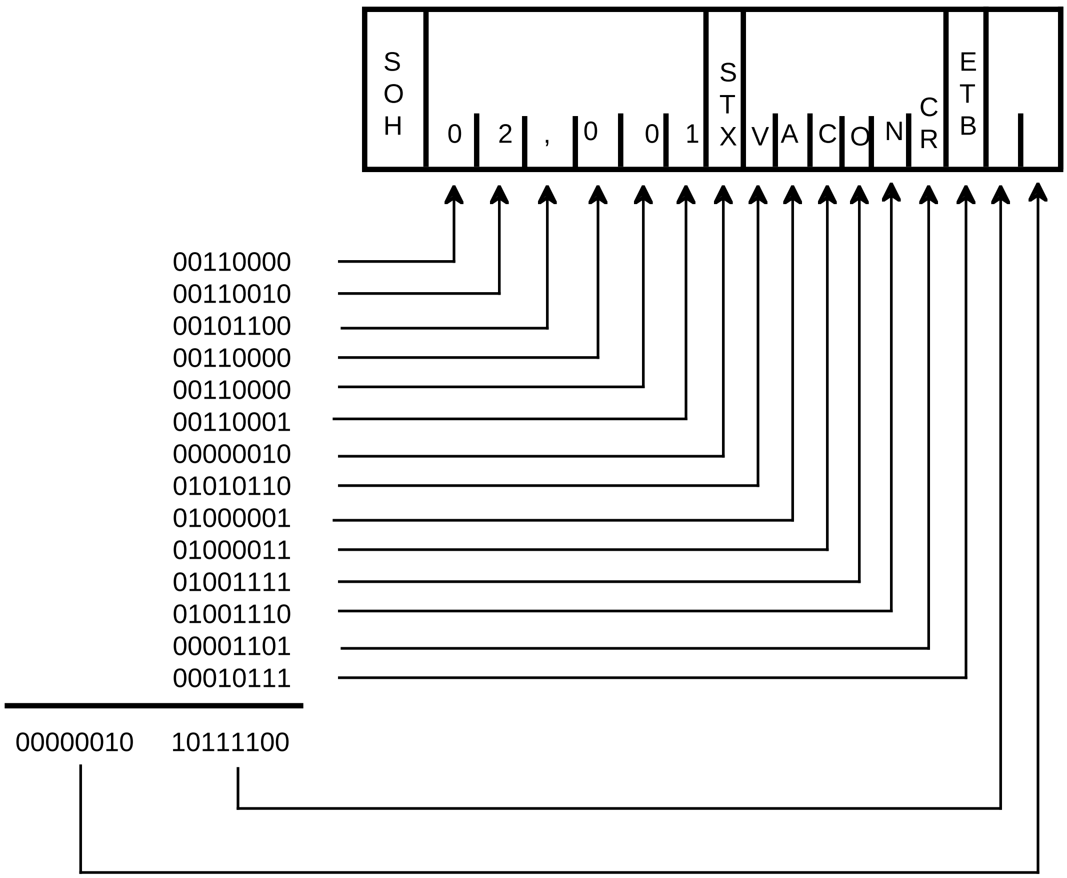


Figure 6.3 Example BCC Calculation

The BCC characters will range from 00000000 to 11111111 (00H to FFH where H represents a hexadecimal value). In this example the BCC calculation results in a transmission sequence of BCH (1011100 binary) as the first BCC character and 02H (00000010 binary) as the second BCC character. These values may or may not be displayable ASCII characters.

6.6 COMMUNICATIONS MONITOR

The ERC Controller features a communication monitoring function to monitor the transmission of data between the ERC and a remote computer. This function is designed to aid in the resolution of communication problems.

The communications monitor is invoked from the ERC operator's panel, using a combination of dedicated and soft keys. To monitor the transmission of data between the ERC and a remote computer the following key sequence is used:

1. Depress the **DISP** (display) key
2 Depress the **Diagnosis** soft key
3. Depress the **-->** soft key until Com Monitor is displayed as a soft key
4. Depress the **Com Monitor** soft key

The monitor on the ERC operator's panel will display the data being transmitted and received by the ERC Controller. Data which is being transmitted from the ERC to the remote computer will be displayed in reverse video. Data being transmitted from the remote computer to the ERC will be displayed in normal video mode. Communication control characters (e.g. ACK0, ENQ) are displayed with < > brackets (e.g. <ACK0>, <ENQ>). BCC (block check characters) are displayed according to the checksum value. These may translate to a non-displayable (not text) ASCII characters.

During the monitoring of transmission data, the **Stop** and **Cancel** soft keys are operational. Depressing the **Stop** soft key halts the monitoring of the transmission data. Depressing the **Cancel** soft key resumes the transmission monitor function.

6.7 ERC OPERATION MODES

6.7.1 Introduction

The ERC Controller has two modes of operation: stand-alone and remote. The ERC Controller cannot operate in remote mode unless the Remote Communications Option is purchased. The purchase of this option allows the ERC Controller to communicate with a remote computer in both stand-alone and remote modes.

When a remote computer is connected to the ERC Controller, the role each device plays is determined by the ERC's operation mode. In stand-alone mode, the ERC Controller is defined as the master (i.e., the controlling device) and the remote computer is defined as the slave (i.e., the device being controlled). In remote mode, the remote computer is defined as the master and the ERC Controller is defined as the slave.

Stand-alone mode provides the capability to communicate with a remote computer for the transfer of various data. Within this mode, the remote computer effectively becomes an external storage device for the ERC Controller.

Remote mode also provides for the transfer of various data, as in stand-alone mode. In addition, when the ERC is in the remote mode of operation, the remote computer is used to control the robot.

6.7.2 Switching Modes

To switch from stand-alone mode to remote mode, press the **REMOTE** key on the ERC operator's panel. The ERC Controller will turn on the remote mode setting signal (#5025) and the LED on the REMOTE key. At the same time, the LEDs of the TEACH and PLAY keys will be turned off.

To cancel remote mode, press either the **TEACH** or **PLAY** keys on the ERC operator's panel. The ERC will turn off the remote mode setting signal (#5025) and the LED on the REMOTE key. The LED on the TEACH or PLAY key will be turned on, depending upon which key was depressed.

Switching between stand-alone and remote mode is also possible by utilizing the remote mode selection signal (#4025) on the I/O Board.

6.8 ERC STAND-ALONE MODE

6.8.1 Introduction

In stand-alone mode, the primary task of the ERC is control of the robot. Additionally, the ERC has the capability to transmit and receive various kinds of data to and from a remote computer. Within this mode, the remote computer effectively becomes an external storage device for the ERC Controller. The following types of data can be transmitted: Job data, Condition data and System data. Both Job and Condition data can be transmitted and received by the ERC. System data may only be transmitted by the ERC for storage on the remote computer.

6.8.2 Communication Command Selection

As stated previously, in stand-alone mode the ERC Controller functions as the master (controlling) device while the remote computer functions as the slave device. In this mode, all communication commands and functions between the ERC and the remote computer are selected through the operator's panel interface on the ERC Controller.

In order to initiate a data communication operation, press the **OP1** key on the ERC operator's panel. Next, press the **Data Com** soft key. The operator's panel CRT will display the data communications function screen. This screen displays the available data types: Job, Condition and System; along with the command functions: Load, Save and Verify.

At this point, use the cursor keys to position the highlighted field on the desired data group type. Once the data group type has been selected, press one of the soft key command functions (**Load**, **Save** or **Verify**).

NOTE: System data can not be loaded from the remote computer to the ERC. System data can only be saved from the ERC to the remote computer.

Following selection of command, the display will change to provide a list of files corresponding to the specified data group type.

6.8.3 Transmission Sequence

This section contains two examples of transmission sequences between the ERC Controller and a remote computer. The first example describes the transmission of a data file from the ERC to a remote computer. The second example describes the converse, the transmission of a data file from the remote computer to the ERC.

The job data file used in examples 6.1 and 6.2 is an independent job called "DEMO". The format (when viewed with a text editor package on a remote computer) and contents of "DEMO" are provided in Figure 6.4, below.

➪ WARNING!

Do not attempt to operate the robot with this job data. The pulse counts may result in unallowable physical motion for your robot. These values may cause damage to the robot or equipment.

```
/JOB
//NAME DEMO
///NPOS 5,0,0,0
///TOOL 0
///PULSE
C000=1234,2345,234,789,9876,1230
C001=123,-2345,234,789,8876,1230
C002=123,1345,234,789,8876,230
C003=2300,12345,789,-200,6635,390
C004=450,12345,89,-200,3691,0
//INST
///DATE 1990/10/20 11:06
///ATTR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
///FRAME BASE
NOP
MOVJ C000 VJ=50.0
MOVL C001 V=100
MOVL C002 V=200
MOVL C003 V=250
MOVJ C004 VJ=70.0
RET
END
```

Figure 6.4 DEMO Job Data Format

6.8.3.1 ERC To Remote Computer

<u>Example 6.1</u> ERC to Remote Computer

The following transactions are used to transmit the contents of "DEMO" from the ERC Controller to the remote computer.

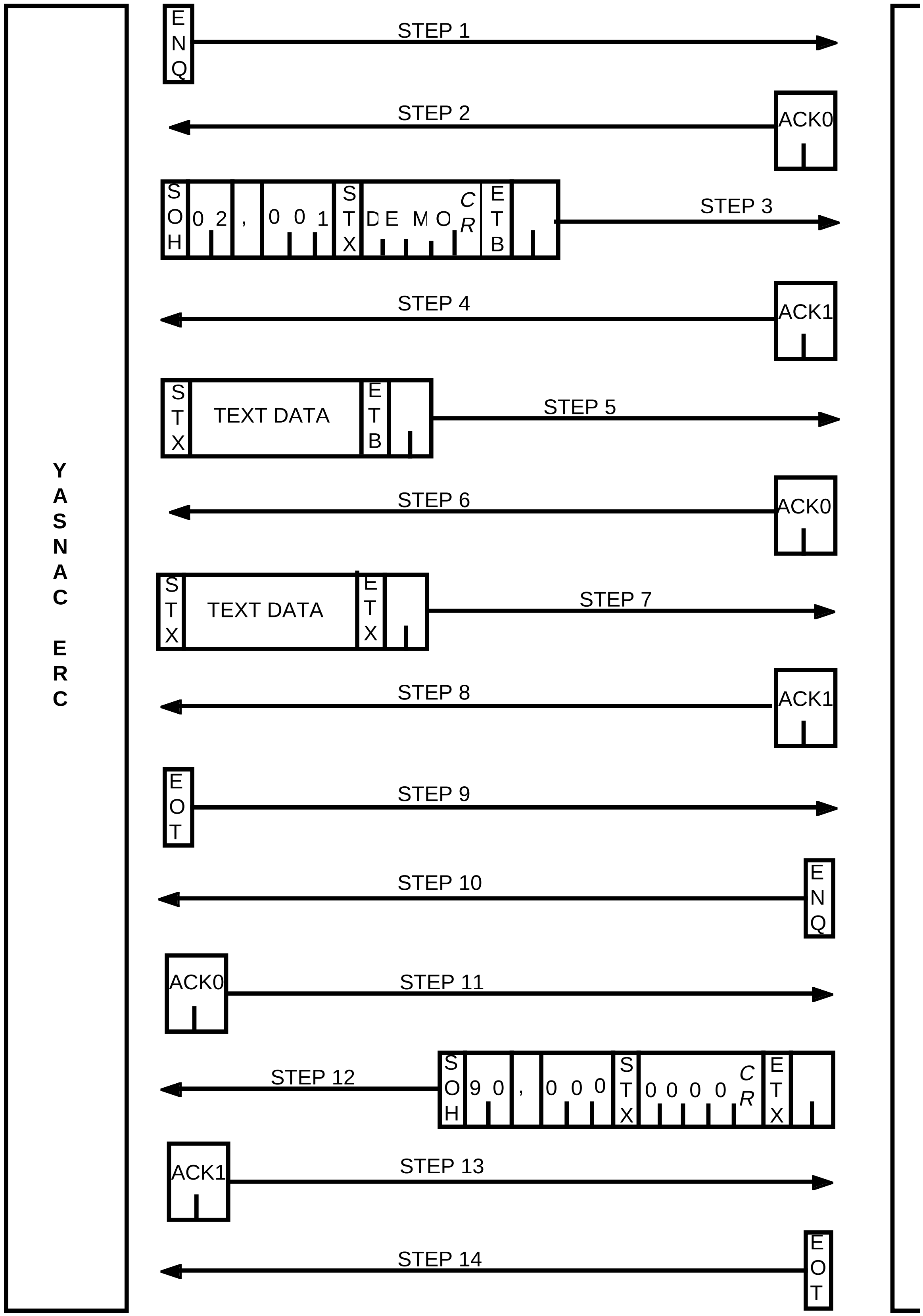
1. The ERC transmits an ENQ to establish the data link with the remote computer.
2. The remote computer responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the ERC transmits the command sequence to save the job data. Header code 02,001 instructs the remote computer to save an independent job file. The file name DEMO is contained within the text field.
4. The remote computer responds with an ACK1.

5.-8. The ERC transmits the file data within the text field of the following transmissions. The number required will depend upon the length of the data file. The remote computer responds with an ACK0 or ACK1 for each transmission.

9. Upon completion of the data file transfer, the ERC issues an EOT. The ERC is in a waiting condition for a response from the remote computer that the file save has been successful.
10. The remote computer transmits an ENQ to establish a data link with the ERC.
11. The ERC responds to the ENQ with an ACK0.
12. After receipt of the ACK0, the remote computer transmits the command sequence indicating successful transfer. Header code 90,000 indicates to the ERC, a response from a remote computer. The response code is contained within the text field.
13. The ERC responds with an ACK1.
14. The remote computer issues an EOT to signal the end of communications.

Figure 6.5 on the following page depicts a transmission sequence, initiated by the ERC, to save an independent job file from the ERC Controller to a remote computer.

ERC ---> REMOTE COMPUTER



The values of the text fields in the transmission of the job file "DEMO" are provided in Figure 6.6, below. The first text block transmits only the file name. The file contents are transmitted by the second and subsequent text blocks. The data is divided every 256 bytes, which is the maximum length of a text block.

CR indicates an ASCII character for carriage return. *LF* indicates an ASCII character for line feed.

1st text block (Step 3 in Example 6.1, Step 8 in Example 6.2). Text length 5 bytes.

DEMO*CR*

2nd text block (Step 5 in Example 6.1, Step 10 in Example 6.2). Text length 256 bytes.

/JOB*CRLF*
//NAME DEMO *CRLF*
///NPOS 5,0,0,0*CRLF*
///TOOL 0*CRLF*
///PULSE*CRLF*
C000=1234,2345,234,789,9876,1230*CRLF*
C001=123,-2345,234,789,8876,1230*CRLF*
C002=123,1345,234,789,8876,230*CRLF*
C003=2300,12345,789,-200,6635,390*CRLF*
C004=450,12345,89,-200,3691,0*CRLF*
//INST*CRLF*
///DATE 1990/10/20 11:06*CRLF*
///AT

3rd text block (Step 7 in Example 6.1, Step 12 in Example 6.2). Text length 155 bytes

TR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*CRLF*
///FRAME BASE *CRLF*
NOP*CRLF*
MOVJ C000 VJ=50.0 *CRLF*
MOVL C001 V=100 *CRLF*
MOVL C002 V=200 *CRLF*
MOVL C003 V=250 *CRLF*
MOVJ C004 VJ=70.0 *CRLF*
RET*CRLF*
END*CRLF*

Figure 6.6 Text Field Values for Examples 6.1 and 6.2

6.8.3.2 Remote Computer To ERC

Example 6.2 Remote Computer to ERC

The following transactions are used to transmit the contents of "DEMO" from the remote computer to the ERC.

1. The ERC transmits an ENQ to establish the data link with the remote computer.
2. The remote computer responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the ERC transmits the command sequence to request the upload of the job data. Header code 02,051 is a request from the ERC, instructing the remote computer to send an independent job file to the ERC. The file name DEMO is contained within the text field.
4. The remote computer responds with an ACK1.
5. The ERC issues an EOT. The ERC then waits for the job data file.
6. The remote computer transmits an ENQ to establish a data link with the ERC.
7. The ERC responds to the ENQ with an ACK0.
8. After receipt of the ACK0, the remote computer transmits the command sequence to upload the job data. Header code 02,001 instructs the ERC to load an independent job file. The file name DEMO is contained within the text field.

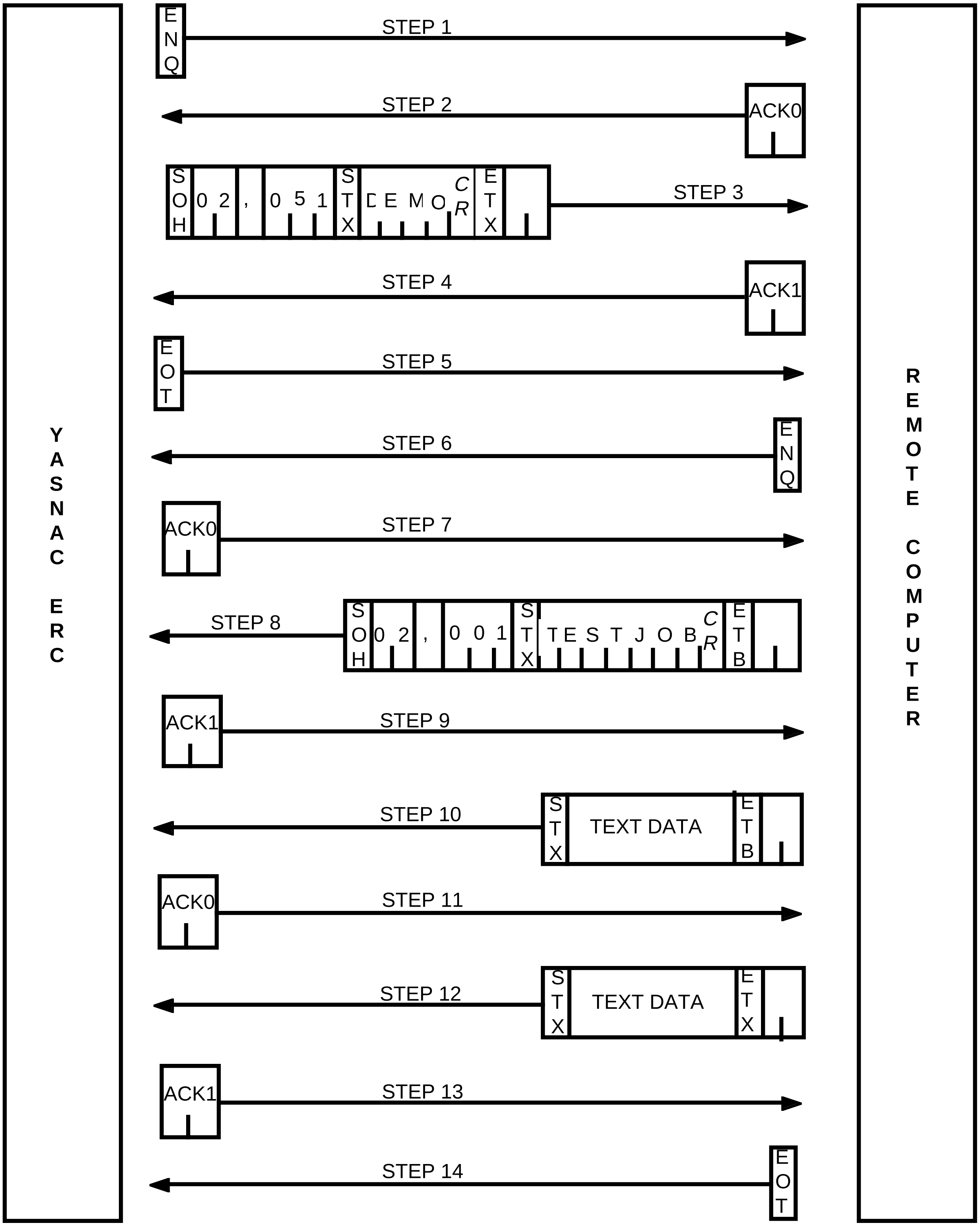
9.-13. The remote computer transmits the file data within the text field of the following transmissions. The number required will depend upon the length of the data file. The ERC responds with an ACK0 or ACK1 for each transmission.

14. Upon completion of the data file transfer, the remote computer issues an EOT.

NOTE: The values of the text field blocks are the same as those given in example 6.1. Refer to Figure 6.6.

Figure 6.7, on the following page, depicts a transmission sequence initiated by the ERC to load an independent job file from a remote computer to the ERC Controller.

ERC <--- REMOTE COMPUTER



6.9 ERC REMOTE MODE

6.9.1 Introduction

In remote mode, the primary task of the ERC is still control of the robot. However, the ERC will receive instructions from a remote computer. In remote mode, the remote computer functions as the master (controlling) device while the ERC Controller functions as the slave device. Within this mode, the remote computer effectively controls the ERC. While the ERC Controller is in remote mode, the remote computer will provide two primary functions: file data transfer and remote robot control.

➪ DANGER!

Improper programming can cause serious personal injury or death and damage to the robot. Check all new programs and/or any changes to programs on the robot in slow speed.

➪ WARNING!

Improper programming can cause unexpected robot motion. Test all programs thoroughly. Only programmers with serial communication and remote machine control experience should be allowed to use the Data Communication Function.

Remote mode provides for the transfer of various data, as in stand-alone mode. However, in remote mode, the transfer of data between the remote computer and the ERC Controller is accomplished in accordance with instructions from the remote computer. This is unlike stand-alone mode, where the ERC controls the transfer of data. The following types of data can be transferred: Job data, Condition data and System data. Both Job and Condition data can be transmitted and received by the ERC. System data may only be transmitted by the ERC for storage on the remote computer.

The second function provided by the remote computer is control of the robot. The remote computer can transmit command functions to be executed by the ERC for robot control and status inspection.

6.9.2 Command Selection

To operate the ERC from a remote computer, press the **REMOTE** key on the ERC operator's panel. The ERC will now be in the remote mode of operation. The remote computer will be able transmit transaction codes consisting of command functions and file data to the ERC Controller. The available commands for file data transfer and robot control are described in the following sections.

6.9.3 Command Validity

The validity of a command, from a remote computer to the ERC Controller, is dependent upon the status of the ERC at the time the command is received. Table 6-5 provides a list of the commands/functions which may be issued by a remote computer. The validity of each command/function is given with respect to the status of the ERC Controller/robot.

Table 6-5 ERC Controller Command Validity

Command Name/Function	Robot Stopped	Robot Running	Panel Hold	Command Hold	Alarm/ Error
CANCEL	Y	Y	-	Y	Y
CYCLE	Y	Y	-	-	Y
DELETE	Y	-	-	-	-*
HOLD	Y	Y	-	Y	Y
HLOCK	Y	Y	Y	Y	Y
JSEQ	Y	-	-	-	-*
JWAIT	Y	Y	-	-	-*
MDSP	Y	Y	Y	Y	Y
RALARM	Y	Y	Y	Y	Y
RESET	Y	Y	-	Y	Y
RJDIR	Y	-	-	-	-*
RJSEQ	Y	Y	Y	Y	Y
RPOS	Y	Y	Y	Y	Y
RPOSJ	Y	Y	Y	Y	Y
RSTATS	Y	Y	Y	Y	Y
SETMJ	Y	-	-	-	-*
START	Y	-	-	-	-*
SVON	Y	Y	-	Y	-*
upload	Y	-	-	-	-*
download	Y	-	-	-	-*

"Y" Indicates those commands/functions which are valid for the ERC's/robot's status.

"-" Indicates those commands/functions which are not valid for the ERC's/robot's status.

"*" Indicates that the ERC Controller will transmit the alarm/error which has occurred.

6.9.4 *ERC Operator's Panel*

When the ERC Controller is in the remote mode of operation, all keys/buttons on the ERC operator's panel are disregarded except for the following:

- DISP
- CANCEL
- Soft Keys
- REMOTE, TEACH, and PLAY Mode Keys
- START and HOLD Buttons
- EMERGENCY STOP Button
- Servo Power ON Button

6.9.5 *Hold/Restart Operation*

The Hold and Restart operations for the ERC Controller can originate from the following sources: Teach Pendant, ERC operator's panel, External input, or a Remote computer. The origin of the Hold operation determines the validity of where a Restart operation can be executed. This section provides a description of valid Hold and Restart operations while the ERC is in the remote mode of operation.

The Hold operation, when executed from the Teach Pendant, ERC operator's panel or External input, is only valid while the robot is in operation (i.e. start lamp is lit). The HOLD command, issued from the remote computer, is valid when the robot is operating or stopped.

Table 6-6 provides a validation check for the origin of a Hold operation with respect to the robot status.

Table 6-6 Hold Operation Validation Check

Robot Status	Teach Pendant Hold	ERC Panel Hold	External Hold	Remote Computer Hold
Stopped	-	-	-	Y
Operating	Y	Y	Y	Y

"Y" Indicates the Hold operation is valid for the robot's status.

"-" Indicates the Hold operation is not valid for the robot's status.

The Restart operation is valid only after a Hold or Emergency Stop operation has occurred.

The Start (Restart) operation is valid only after the robot has been stopped by a Hold operation (Teach Pendant, External Hold, or ERC operator's panel) during the robot's operation or an Emergency Stop (Teach Pendant, External Emergency Stop, or ERC operator's panel). The START command, issued by a remote computer, is not valid if the robot has been stopped by a Hold or Emergency Stop operation from a source other than the remote computer. The Start /Restart operation (Teach Pendant, External Start, ERC operator's panel) is not valid if the robot has been stopped by a HOLD command issued from the remote computer. Table 6-7, below, provides a validation check for the origin of a Start/Restart operation with respect to the robot stopping factor.

Table 6-7 Start/Restart Validation Check

Stop Factor	ERC Panel	External	Remote Computer
Teach Pendant Hold	Y	-	-
ERC Panel Hold	Y	-	-
External Input Hold	-	Y	-
Remote Computer HOLD	-	-	Y
Teach Pendant E-Stop	Y	-	-
ERC Panel Emergency Stop	Y	-	-
External Emergency Stop	-	Y	-
Remote computer Servo Off	-	-	Y
Cycle Step Stop	-	-	Y

"Y" Indicates the Restart operation is valid from that particular origin for that particular factor which stopped the robot.

"-" Indicates the Restart operation is not valid from that particular origin for that particular factor which stopped the robot.

6.9.6 Servo Power On/Off

In ERC remote operation mode, switching the servo power on and off is normally accomplished by the SVON command which is issued by the remote computer. After an Emergency Stop, the servo power can only be turned on from through the device which issued the Emergency Stop.

NOTE: The Teach Pendant cannot turn the servo power on.

Table 6-8, below, provides a validation check for the origin of the Servo On operation with respect to the condition which turned the Servo Off .

Table 6-8 Servo On Validation Check

Servo Off Condition	ERC Panel	External	Remote Computer
Teach Pendant E-Stop	Y	-	-
ERC Panel Emergency Stop	Y	-	-
External Emergency Stop	-	Y	-
Remote computer Servo Off	-	-	Y

"Y" Indicates the Servo Power ON operation is valid from that particular origin and for that particular factor which turned the Servo Power Off.

"-" Indicates the Servo Power ON operation is not valid from that particular origin and for that particular factor which turned the Servo Power Off.

6.9.7 Data File Transfer Function

This section contains two examples of transmission sequences between a remote computer and the ERC Controller. The first example describes the transmission of a data file from the ERC to a remote computer. The second example describes the converse, the transmission of a data file from the remote computer to the ERC. Since the ERC is operating in remote mode, the remote computer initiates the file data transfers.

The job data file used in examples 6.3 and 6.4 is an independent job called "DEMO". The format (when viewed with a text editor package on a remote computer) and contents of "DEMO" are provided in Figure 6.8.

➪ *WARNING!*

Do not attempt to operate the robot with this job data. The pulse counts may result in unallowable physical motion for your robot. These values may cause damage to the robot or equipment.

```
/JOB
//NAME DEMO
///NPOS 5,0,0,0
///TOOL 0
///PULSE
C000=1234,2345,234,789,9876,1230
C001=123,-2345,234,789,8876,1230
C002=123,1345,234,789,8876,230
C003=2300,12345,789,-200,6635,390
C004=450,12345,89,-200,3691,0
//INST
///DATE 1990/10/20 11:06
///ATTR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
///FRAME BASE
NOP
MOVJ C000 VJ=50.0
MOVL C001 V=100
MOVL C002 V=200
MOVL C003 V=250
MOVJ C004 VJ=70.0
RET
END
```

Figure 6.8 DEMO Job Data File

6.9.7.1 ERC To Remote Computer Example

Example 6.3 ERC to Remote Computer

The following transactions are used to transmit the contents of "DEMO" from the ERC Controller to the remote computer.

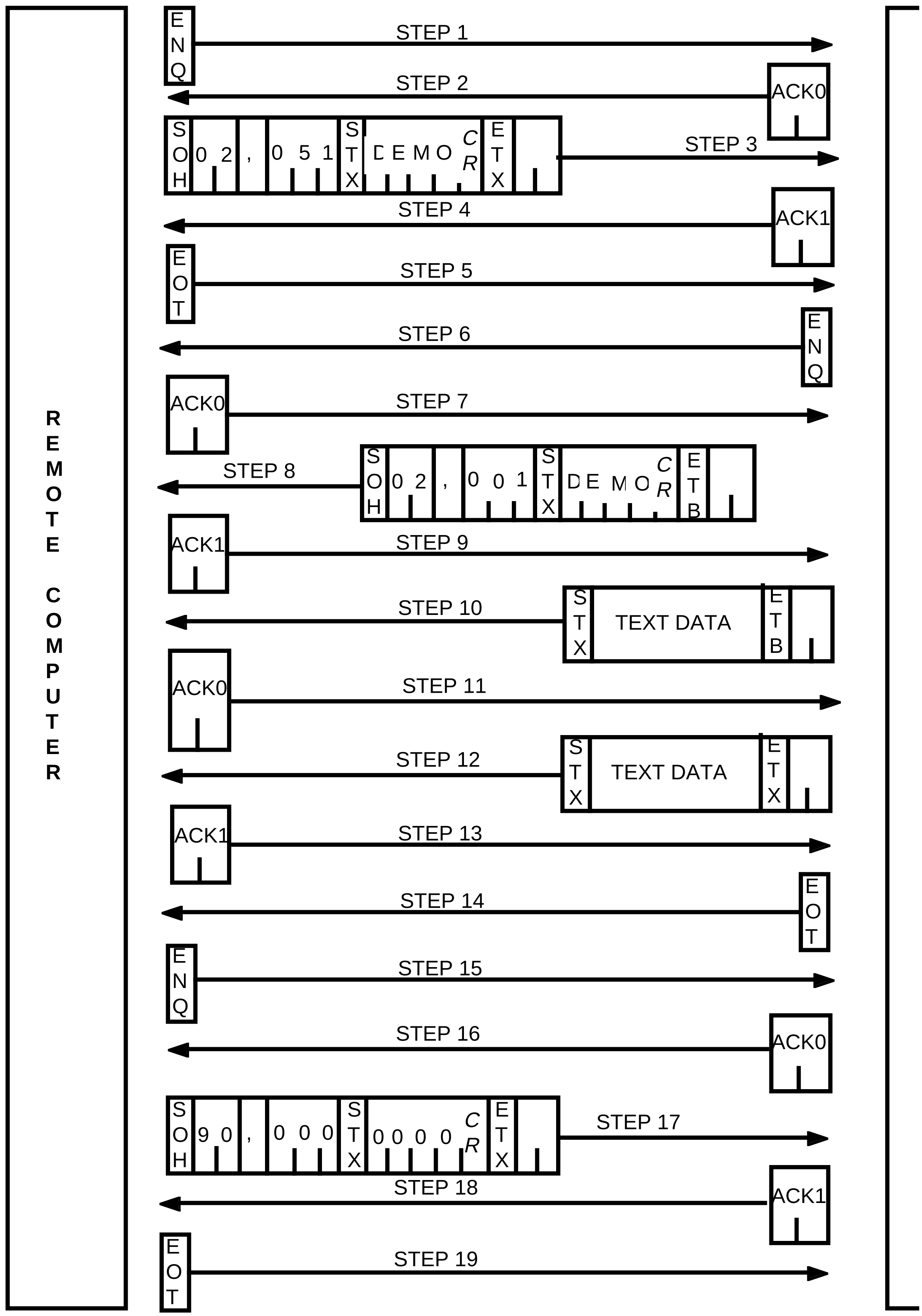
1. The remote computer transmits an ENQ to establish the data link with the ERC.
2. The ERC responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the remote computer transmits the command sequence to request the download (save) of the job data. Header code 02,051 is a request from the remote computer instructing the ERC to send an independent job file to the remote computer. The file name DEMO is contained within the text field.
4. The ERC responds with an ACK1.
5. The remote computer issues an EOT. The remote computer then waits for the job data file.
6. The ERC transmits an ENQ to establish a data link with the remote computer.
7. The remote computer responds to the ENQ with an ACK0.
8. After receipt of the ACK0, the ERC transmits the command sequence to save the job data. Header code 02,001 instructs the remote computer to save an independent job file. The file name DEMO is contained within the text field.

9.-13. The ERC transmits the file data within the text field of the following transmissions. The number required will depend upon the length of the data file. The remote computer responds with an ACK0 or ACK1 for each transmission.

14. Upon completion of the data file transfer, the ERC issues an EOT. The ERC is in a waiting condition for a response from the remote computer telling it that the file save has been successful.
15. The remote computer transmits an ENQ to establish a data link with the ERC.
16. The ERC responds to the ENQ with an ACK0.
17. After receipt of the ACK0, the remote computer transmits the command sequence indicating successful transfer. Header code 90,000 is a response from a remote computer to the ERC. The response code is contained within the text field.
18. The ERC responds with an ACK1.
19. The remote computer issues an EOT to signal the end of communications.

Figure 6.9, on the following page, depicts a transmission sequence, initiated by the remote computer, to save an independent job file from the ERC Controller to a remote computer.

REMOTE COMPUTER <--- ERC



The values of the text fields in the transmission of the job file "DEMO" are provided in Figure 6.10. The first text block transmits only the file name. The file contents are transmitted by the second and subsequent text blocks. The data is divided every 256 bytes, which is the maximum length of a text block.

CR indicates an ASCII character for carriage return. *LF* indicates an ASCII character for line feed.

1st text block (Step 8 in Example 6.3, Step 3 in Example 6.4) Text length 5 bytes.

DEMO*CR*

2nd text block (Step 10 in Example 6.3, Step 5 in Example 6.4) Text length 256 bytes.

/JOB*CRLF*
//NAME DEMO*CRLF*
///NPOS 5,0,0,0*CRLF*
///TOOL 0*CRLF*
///PULSE*CRLF*
C000=1234,2345,234,789,9876,1230*CRLF*
C001=123,-2345,234,789,8876,1230*CRLF*
C002=123,1345,234,789,8876,230*CRLF*
C003=2300,12345,789,-200,6635,390*CRLF*
C004=450,12345,89,-200,3691,0*CRLF*
//INST*CRLF*
///DATE 1990/10/20 11:06*CRLF*
///AT

3rd text block (Step 12 in Example 6.3, Step 7 in Example 6.4) Text length 155 bytes

TR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0*CRLF*
///FRAME BASE*CRLF*
NOP*CRLF*
MOVJ C000 VJ=50.0*CRLF*
MOVL C001 V=100*CRLF*
MOVL C002 V=200*CRLF*
MOVL C003 V=250*CRLF*
MOVJ C004 VJ=70.0*CRLF*
RET*CRLF*
END*CRLF*

Figure 6.10 Text Field Values for Examples 6.3 and 6.4

6.9.7.2 Remote Computer To ERC

Example 6.4 Remote Computer to ERC

The following transactions are used to transmit the contents of "DEMO" from the remote computer to the ERC Controller.

1. The remote computer transmits an ENQ to establish the data link with the ERC.
2. The ERC responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the remote computer transmits the command sequence to load the job data. Header code 02,001 instructs the ERC to load an independent job file. The file name DEMO is contained within the text field.
4. The ERC responds with an ACK1.
5. The remote computer transmits the file data within the text field of the following transmissions. The number required will depend upon the length of the data file.
6. The ERC responds with an ACK0 or ACK1 for each transmission.
7. Upon completion of the data file transfer, the remote computer issues an EOT.

NOTE: The values of the text field blocks are the same as those given in Example 6.3. Refer to Figure 6.10.

Figure 6.11, on the following page, depicts a transmission sequence, initiated by the remote computer, to load an independent job file from a remote computer to the ERC Controller.

REMOTE COMPUTER ---> ERC

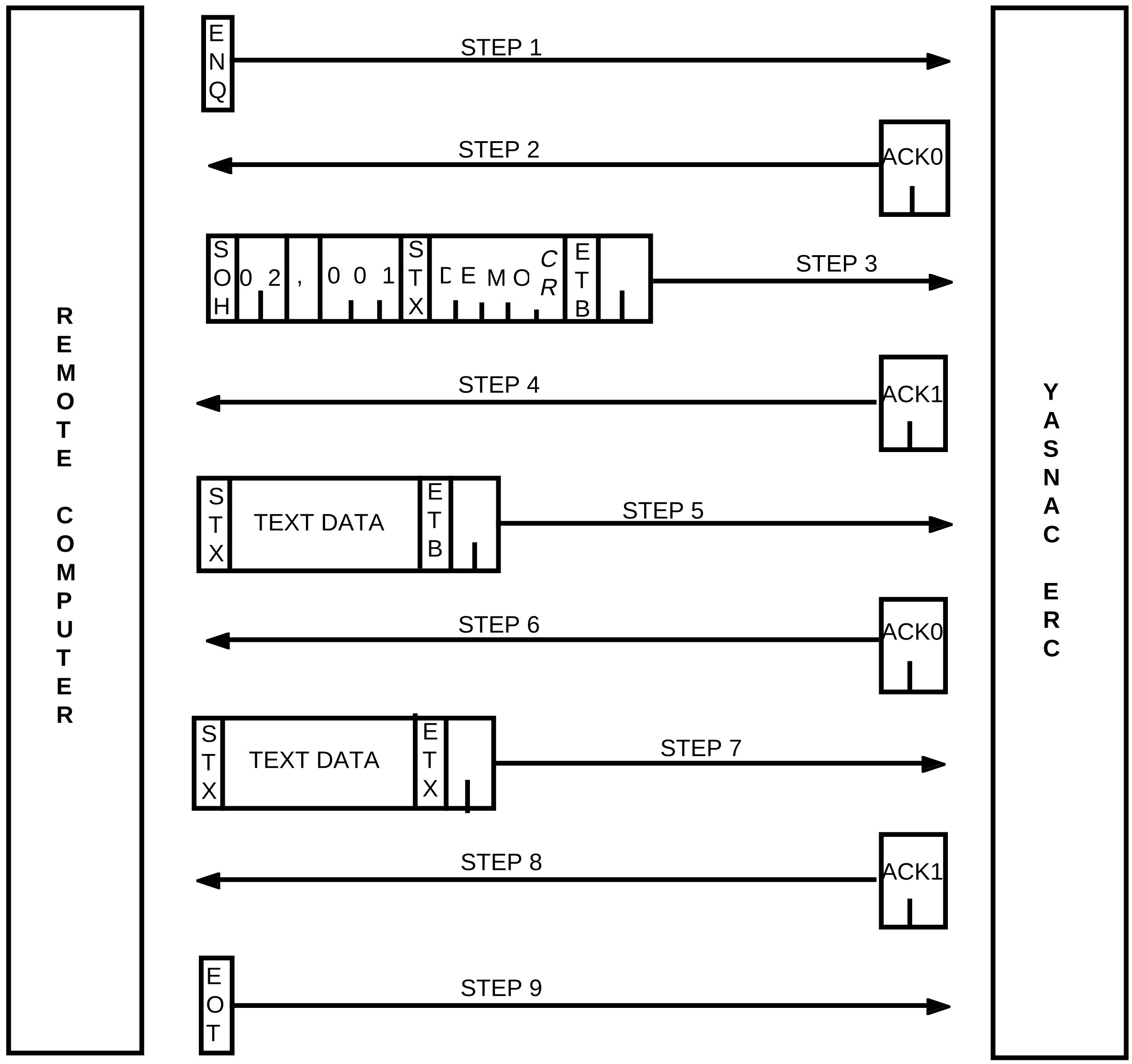


Figure 6.11 Transmission Sequence for Example 6.4

6.9.8 Robot Control Function

The robot control functions, which are available in remote mode, can be divided into two (2) groups: System Control functions and Status Read functions.

6.9.8.1 System Control Functions

System control functions are used to control the robot operations. A list of the system control commands along with their corresponding function is provided in Table 6-9 below.

Table 6-9 System Control Functions

Command	Robot Control Function
CANCEL	Cancels the error status.
CYCLE	Selects a motion cycle mode.
DELETE	Deletes the specified job.
HLOCK	Turns operator's panel interlock on or off.
HOLD	Halts the robot manipulator.
JSEQ	Specifies a job and line number.
JWAIT	Checks robot operation status.
MDSP	Displays message on the operator's panel CRT.
RESET	Resets the robot alarms.
SETMJ	Assigns the specified job as the master job.
START	Starts robot operation.
SVON	Turns servo power on or off.

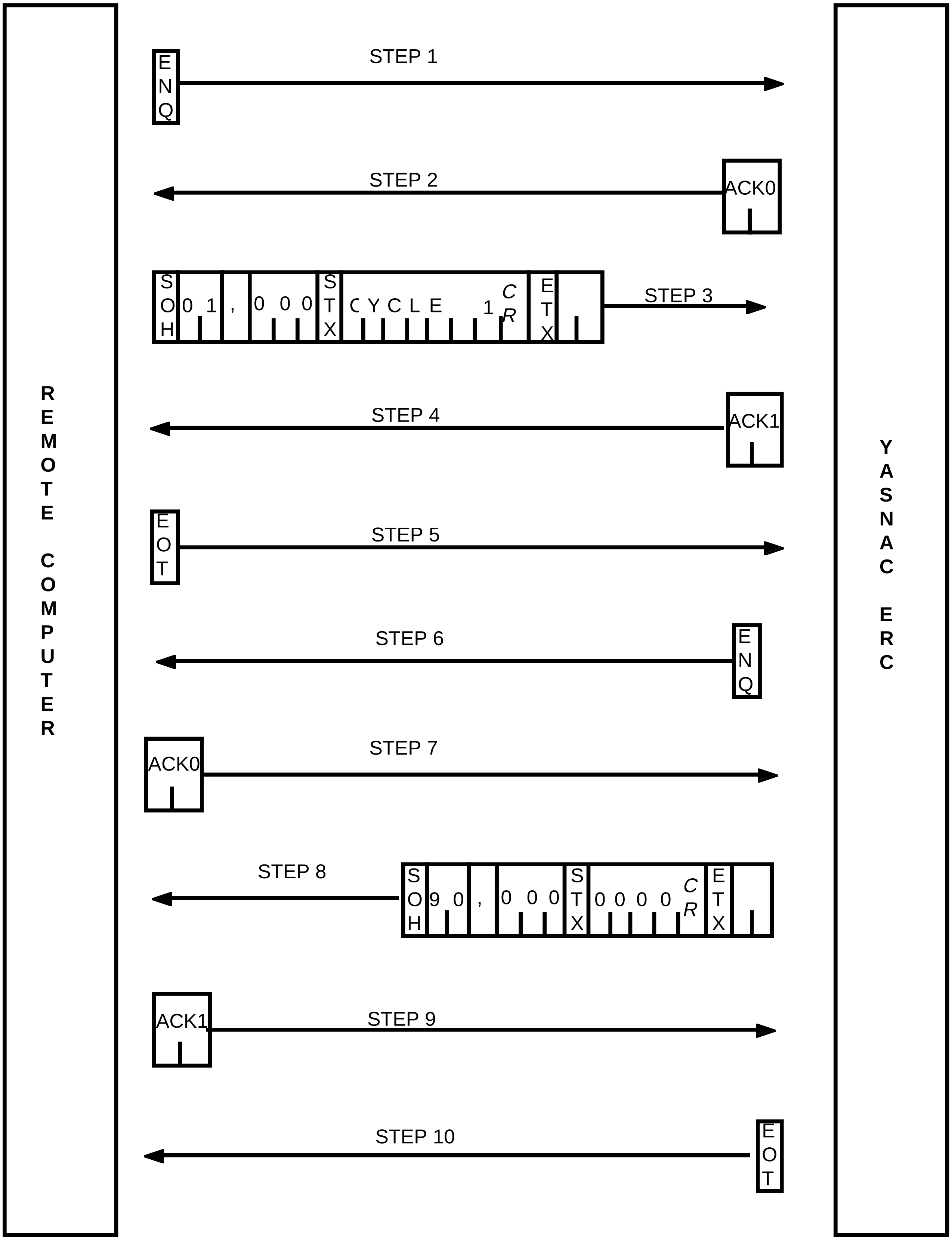
A detailed description of each system control function is provided below. The description details the command name, function, format, data, response and an example. Note that each command and each response contains a carriage return control character denoted by *CR*. This control character is required at the end of the data field in each system control command and response code. Commands which pass data to the ERC Control must contain one "space" character between the command name and the data that is being sent as part of the command (e.g. HOLD, SVON, etc.). If more than one data item is specified in a command, commas must separate the data items (e.g. JSEQ). "Error code" is a four (4) character integer code. All commands receive a response from the ERC Controller. The response type is indicated by the header number and sub code values (refer to Table 6-10). Responses which contain data items in the text field (header code 90, sub code 001) will have the data items separated by commas. There are no spaces between data items.

An example of the transaction between the ERC Controller and the remote computer is provided for the CYCLE command. The following transactions are used to execute the CYCLE command from the remote computer to the ERC Controller.

1. The remote computer transmits an ENQ to establish the data link with the ERC.
2. The ERC responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the remote computer transmits the command sequence to execute the CYCLE command. Header code 01,000 instructs the ERC to execute a command from a remote computer. The command name CYCLE along with a "space" character, a 1, and a CR are contained within the text field.
4. The ERC responds to the command with an ACK1.
5. The remote computer issues an EOT. It then waits for confirmation of the command data.
6. The ERC transmits an ENQ to establish a data link with the remote computer.
7. The remote computer responds to the ENQ with an ACK0.
8. After receipt of the ACK0, the ERC transmits the command sequence indicating a successful transfer. Header code 90,000 is a response from the ERC to the remote computer. The response is contained within the text field.
9. The remote computer responds with an ACK1.
10. The ERC issues an EOT to signal the end of communications.

Figure 6.12, on the following page, depicts a transmission sequence, initiated by the remote computer, to execute the CYCLE command and put the ERC Controller in "step" mode.

CYCLE Command



6.9.8.1.1 CANCEL

Command Name:	CANCEL
Command Function:	Cancels the error status.
Command Format:	CANCEL*CR*
Command Data:	None
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	CANCEL*CR*
Example Response:	0000*CR*
Remarks:	None

6.9.8.1.2 CYCLE

Command Name:	CYCLE
Command Function:	Selects a motion cycle mode.
Command Format:	CYCLE "command data"*CR*
Command Data:	"1" assigns step by step mode; "2" assigns a one (1) cycle mode; "3" assigns an automatic repeating mode.
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	CYCLE 3*CR*
Example Response:	0000*CR*
Remarks:	None

6.9.8.1.3 DELETE

Command Name:	DELETE
Command Function:	Deletes the specified job.
Command Format:	DELETE "command data"*CR*
Command Data:	Job name or "*"
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	DELETE **CR*
Example Response:	0000*CR*
Remarks:	The "*" (asterisk) is used to delete all jobs in the ERC.

6.9.8.1.4 HLOCK

Command Name:	HLOCK
Command Function:	Turns operator's panel interlock on or off.
Command Format:	HLOCK "command data"*CR*
Command Data:	"0" turns interlock off; "1" turns interlock on
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	HLOCK 1*CR*
Example Response:	0000*CR*
Remarks:	When the operator's panel is interlocked, only the following keys are functional: START HOLD EMERGENCY STOP SERVO ON

6.9.8.1.5 HOLD

Command Name:	HOLD
Command Function:	Halts the robot manipulator.
Command Format:	HOLD "command data"*CR*
Command Data:	"0" turns hold off, "1" turns hold on
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	HOLD 1*CR*
Example Response:	0000*CR*
Remarks:	None

6.9.8.1.6 JSEQ

Command Name:	JSEQ
Command Function:	Specifies a job and line number.
Command Format:	JSEQ "command data 1, command data 2"*CR*
Command Data:	Job name, line number (0 to 9999)
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	JSEQ TESTJOB, 10*CR*
Example Response:	0000*CR*
Remarks:	None

6.9.8.1.7 JWAIT

Command Name:	JWAIT
Command Function:	Waits the specified time before returning the robot's job status (completed or incomplete).
Command Format:	JWAIT "command data"*CR*
Command Data:	"-1" waits infinitely; "0" to "32767" wait time in seconds
ERC Response:	0*CR* for complete; -1*CR* for incomplete;
or	"error code"*CR*
Example Command:	JWAIT 10*CR*
Example Response:	0000*CR*
Remarks:	The response does not occur until operation is completed or wait time expires. The following factors result in a returned value of incomplete: 1. When robot stops by panel, Teach Pendant, or external hold. 2. When robot stops by an alarm or error. 3. When robot stops by EMERGENCY STOP. 4. When specified wait time is up.

6.9.8.1.8 MDSP

Command Name:	MDSP
Command Function:	Displays the message data on the CRT of the ERC's operator panel.
Command Format:	MDSP "command data"*CR*
Command Data:	ASCII characters. The maximum display is twenty-eight (28) characters.
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	MDSP In continuous operation*CR*
Example Response:	0000*CR*
Remarks:	None

6.9.8.1.9 RESET

Command Name:	RESET
Command Function:	Resets the robot alarms.
Command Format:	RESET*CR*
Command Data:	None
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	RESET*CR*
Example Response:	0000*CR*
Remarks:	The transmission alarm can only be reset from the operator's panel on the ERC.

6.9.8.1.10 SETMJ

Command Name:	SETMJ
Command Function:	Assigns the specified job as the master job.
Command Format:	SETMJ "command data"*CR*
Command Data:	Job name
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	SETMJ TESTJOB*CR*
Example Response:	0000*CR*
Remarks:	The specified job is also set as the job for execution.

6.9.8.1.11 START

Command Name:	START
Command Function:	Starts robot operation.
Command Format:	START "command data"*CR*
Command Data:	Job name or none
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	START TESTJOB*CR*
Example Response:	0000*CR*
Remarks:	The specified job name becomes the master job. The operation starts at the beginning of the job. If no job name is specified, the operation starts from the line number the currently active job assigned.

6.9.8.1.12 SVON

Command Name:	SVON
Command Function:	Turns servo power on or off.
Command Format:	SVON "command data"*CR*
Command Data:	"0" turns servo power off; "1" turns servo power on
ERC Response:	0000*CR* or "error code"*CR*
Example Command:	SVON 1*CR*
Example Response:	0000*CR*
Remarks:	None

6.9.8.2 Status Read Functions

Status read functions are provided to inquire upon the status of information internal to the ERC Controller. A list of the status read commands along with their corresponding function is provided below.

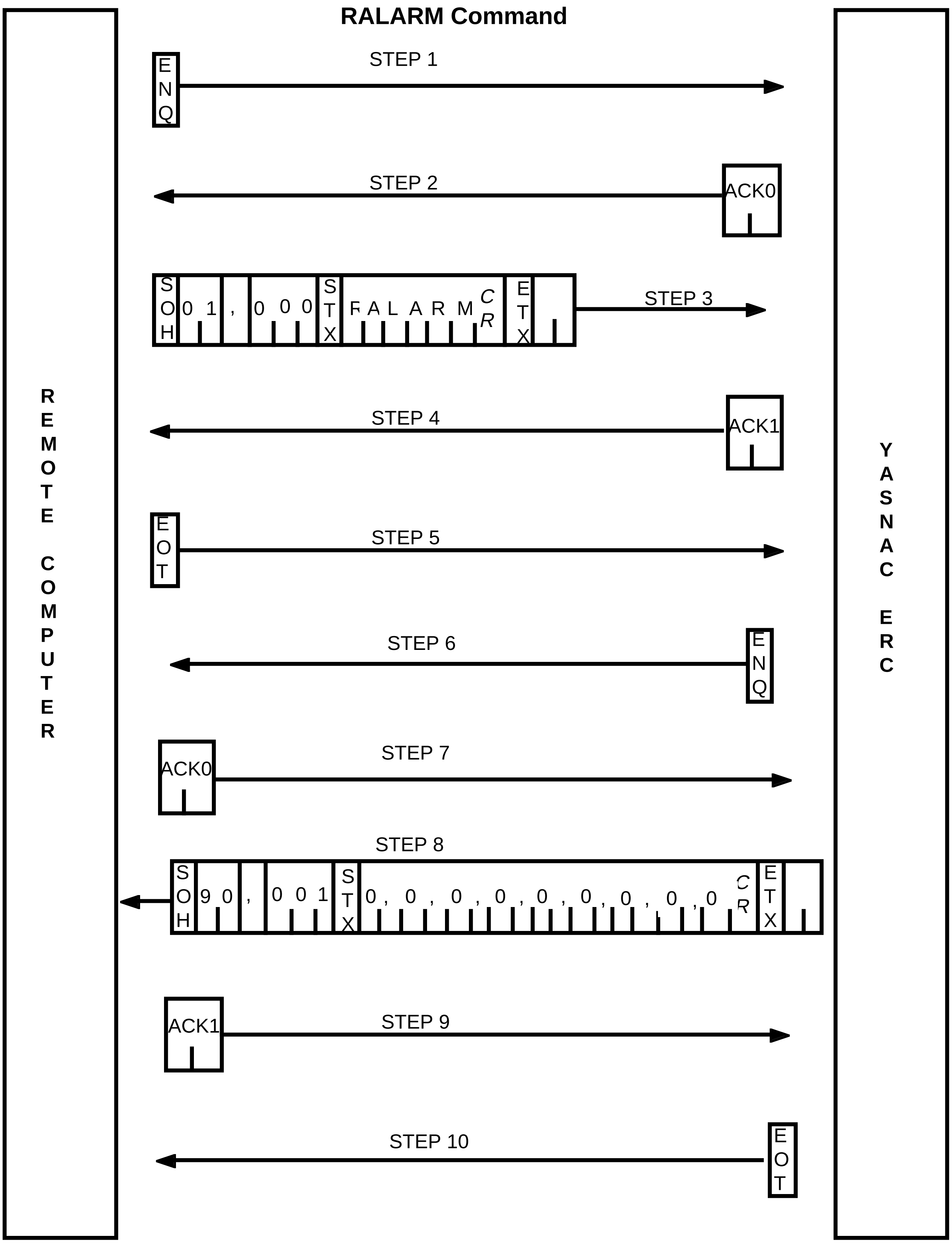
Command	Robot Control Function
RALARM	Reads error and alarm codes.
RJDIR	Reads all job names or names associated with a master job.
RJSEQ	Reads the current job name, line number and step number.
RPOS	Reads the current robot position in rectangular coordinates.
RPOSJ	Reads the current robot position in joint coordinates.
RSTATS	Reads the status of several robot conditions.

A detailed description of each status read function is provided within this section. The description details the command name, function, format, data, response and an example. Note that each command and each response contains a carriage return control character denoted by *CR*. This control character is required at the end of the data field in each system control command and response code. Commands which pass data to the ERC Control must contain one "space" character between the command name and the data that is being sent as part of the command (e.g. RJDIR). "Error code" is a four (4) character integer code. All commands receive a response from the ERC Controller. The response type is indicated by the header number and sub code values (refer to Table 6-10). Responses which contain data items in the text field (header code 90, sub code 001) will have the data items separated by commas. There are no spaces between data items.

An example of the transaction between the ERC Controller and the remote computer is provided for the RALARM command. The following transactions are used to execute the CYCLE command from the remote computer to the ERC Controller.

1. The remote computer transmits an ENQ to establish the data link with the ERC.
2. The ERC responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the remote computer transmits the command sequence to execute the RALARM command. Header code 01,000 instructs the ERC to execute a command from a remote computer. The command name RALARM and a CR are contained within the text field.
4. The ERC responds with an ACK1.
5. The remote computer issues an EOT. It then waits for confirmation of the command data.
6. The ERC transmits an ENQ to establish a data link with the remote computer.
7. The remote computer responds to the ENQ with an ACK0.
8. After receipt of the ACK0, the ERC transmits the command sequence indicating a successful transfer. Header code 90,001 is a response from the ERC to the remote computer, indicating the transmission of data in the text field. The response data is contained within the text field. For the RALARM command nine (9) data items are returned, each data item is separated by a comma. A CR indicates data termination within the text field.
9. The remote computer responds with an ACK1.
10. The ERC issues an EOT to signal the end of communications.

Figure 6.13, on the following page, depicts a transmission sequence, initiated by the remote computer, to execute the RALARM command to request a listing of alarm and error codes.


RALARM Command
REMOTE COMPUTER
YASNAC ERC
STEP 1
ENQ
STEP 2
ACK0
STEP 3
SOH 0 1 , 0 0 0 STX RALARM CR ETX
STEP 4
ACK1
STEP 5
EOT
STEP 6
ENQ
STEP 7
ACK0
STEP 8
SOH 9 0 , 0 0 1 STX 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 CR ETX
STEP 9
ACK1
STEP 10
EOT

6.9.8.2.1 RALARM

Command Name:	RALARM
Command Function:	Reads error and alarm codes.
Command Format:	RALARM*CR*
Command Data:	None
ERC Response:	Data(1), Data(2), ..., Data(9)*CR* or "error code"*CR*
Example Command:	RALARM*CR*
Example Response:	0,1234,12,0,0,0,0,0,0*CR*
Remarks:	The response data is in the following format:
	Data(1): Error code (0 to 9999)
	Data(2): Alarm code (0 to 9999)
	Data(3): Alarm data (0 to 256)
	Data(4): Alarm code (0 to 9999)
	Data(5): Alarm data (0 to 256)
	Data(6): Alarm code (0 to 9999)
	Data(7): Alarm data (0 to 256)
	Data 8): Alarm code (0 to 9999)
	Data(9): Alarm data (0 to 256)

6.9.8.2.2 RJDIR

Command Name:	RJDIR
Command Function:	Reads all job names or job names connected with a master job
Command Format:	RJDIR "command data"*CR*
Command Data:	Job name (for a specific master job) or "*" (for all jobs)
ERC Response:	Data(1),Data(2), ...Data(n) *CR* or "error code"*CR*
Example Command:	RJDIR MASTER*CR*
Example Response:	WORK-A,WORK-B,SAMPLE-*CR*
Remarks:	The response data is in the following format:
	Data(1): Job name 1
	Data(2): Job name 2
	•
	•
	•
	Data(n): Job name n

6.9.8.2.3 RJSEQ

Command Name:	RJSEQ
Command Function:	Reads the current job name, line number and step number.
Command Format:	RJSEQ*CR*
Command Data:	None
ERC Response:	Data(1),Data(2),Data(3)*CR* or "error code"*CR*
Example Command:	RJSEQ*CR*
Example Response:	TESTJOB,10,5*CR*
Remarks:	The response data is in the following format:
	Data(1): Current job name
	Data(2): Current line number (0 to 9999)
	Data(3): Current step number (0 to 99)

6.9.8.2.4 RPOS

Command Name:	RPOS
Command Function:	Reads the current robot position data in rectangular coordinates.
Command Format:	RPOS*CR*
Command Data:	None
ERC Response:	Data(1),Data(2), ...,Data(15)*CR* or "error code"*CR*
Example Command:	RPOS*CR*
Example Response:	1205.1,50.34,712.3,159.2,12.35,25.3, 0,0,0,0,0,0,0,0,0*CR*
Remarks:	The response data is in the following format:
	Data(1): X axis (mm)
	Data(2): Y axis (mm)
	Data(3): Z axis (mm)
	Data(4): List angle TX (°)
	Data(5): List angle TY (°)
	Data(6): List angle TZ (°)
	Data(7): Type 1 ("0" flip; "1" no flip)
	Data(8): Type 2 ("0" upper arm; "1" lower arm)
	Data(9): Type 3 ("0" front; "1" back)

Data(10): Pulse number of 7th axis
(Traverse axis indicated in mm)
Data(11): Pulse number of 8th axis
(Traverse axis indicated in mm)
Data(12): Pulse number of 9th axis
(Traverse axis indicated in mm)
Data(13): Pulse number of 10th axis
Data(14): Pulse number of 11th axis
Data(15): Pulse number of 12th axis

6.9.8.2.5 RPOSJ

Command Name:	RPOSJ
Command Function:	Reads the current robot position data in joint coordinates.
Command Format:	RPOSJ*CR*
Command Data:	None
ERC Response:	Data (1),Data(2), ...,Data(12) *CR* or "error code"*CR*
Example Command:	RPOSJ*CR*
Example Response:	500,2600,1250,10789,624,36, 0,0,0,0,0,0*CR*
Remarks:	The response data is in the following format:

Data(1): Pulse number of S axis
Data(2): Pulse number of L axis
Data(3): Pulse number of U axis
Data(4): Pulse number of R axis
Data(5): Pulse number of B axis
Data(6): Pulse number of T axis
Data(7): Pulse number of 7th axis
Data(8): Pulse number of 8th axis
Data(9): Pulse number of 9th axis
Data(10): Pulse number of 10th axis
Data(11): Pulse number of 11th axis
Data(12): Pulse number of 12th axis

6.9.8.2.6 RSTATS

Command Name:	RSTATS
Command Function:	Reads the status of cycle operation, alarm error and servo.

Command Format:	RSTATS*CR*
Command Data:	None
ERC Response:	Data(1),Data(2)*CR* or "error code"*CR*
Example Command:	RSTATS*CR*
Example Response:	12,64*CR*
Remarks:	The response data is in the following format:

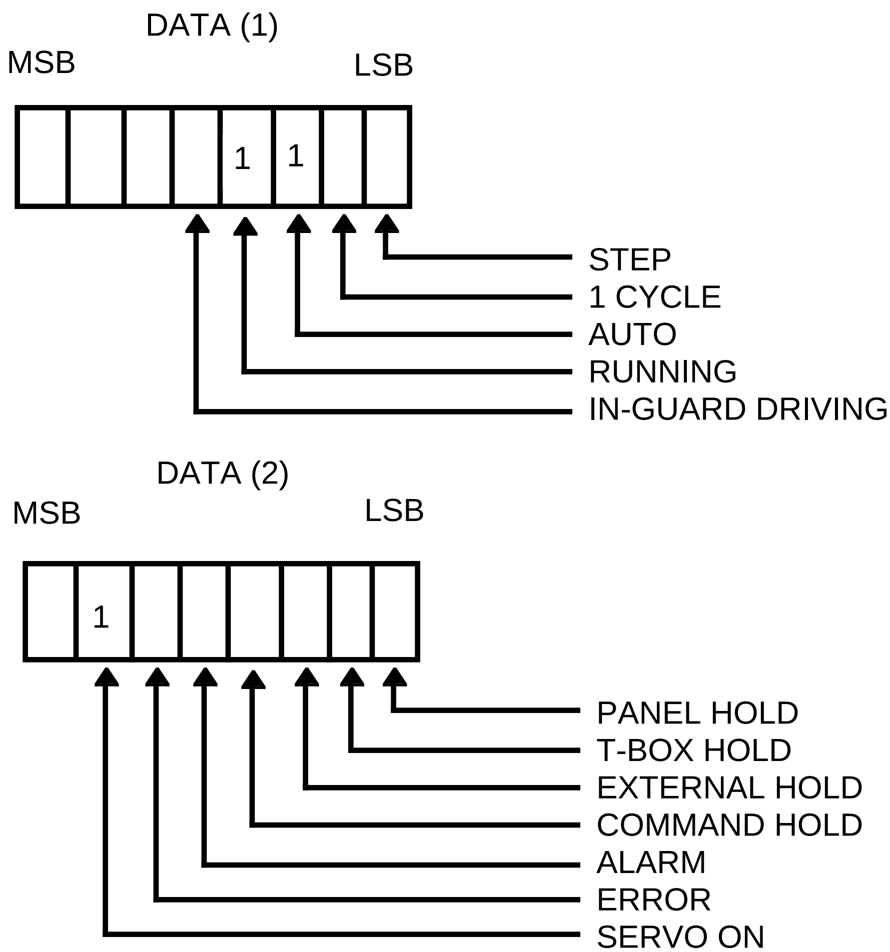


Figure 6.14 Response Data Format for RSTATS Command

6.10 DATA COMMUNICATION WITH DCI FUNCTION

6.10.1 Introduction

The Data Communications with DCI function provides the capability to automatically transfer job and variable data between the ERC controller and a remote computer through the use of INFORM programming instructions.

The Data Communications with DCI function is only valid when the ERC Controller is operating in standalone mode. An alarm will occur if the instructions are attempted while the ERC Controller is operating in remote mode.

6.10.2 Command Selection

Two functions are provided by the Data Communications with DCI:

- Load/Save/Delete of Job data files
- Load/Save of variable data

To utilize the Data Communications with DCI functions capability, INFORM language commands are programmed within the Job data file.

6.10.3 Load/Save/Delete Function for Jobs

Job data files can be loaded to the ERC Controller from a remote computer, and saved from the ERC Controller to a remote computer. There are two (2) methods of specification for job names.

- Job names can be specified directly.
- Job names can be specified by pattern input through the input group feature.

Additionally, Job data files can be deleted from the ERC Controller.

A description of the Load, Save and Delete commands is provided in the following three sections. The description details the command name, function, format and an example. Required command components are depicted in capital letters. The "job name" would be replaced with the actual name of a valid job. "Job type extension" would be replaced by either JBR for a related (master) job or JBI for an independent job. "Input group number" would be replaced with the number of the desired input group. The conditional data, represented by (IF syntax), can contain a valid INFORM programming language conditional statement. The conditional data statement is optional in these commands. Refer to the "K" SERIES ROBOT OPERATOR'S MANUAL for a description of the INFORM programming language and conditional statements.

6.10.3.1 LOADJ

Function:

The LOADJ (load job) command specifies a single job or related jobs to be loaded from the remote computer to the ERC Controller.

Format:

For direct specification of the job name.

LOADJ JOB:job name job type extension (IF syntax)

For specification of job names through the pattern input of the input group.

LOADJ IG#input group number job type extension (IF syntax)

Example:

For direct specification of the job name.

LOADJ JOB:123 JBR IF IN#14=1

For specification of job names through the pattern input of the input group.

LOADJ IG#01 JBI IF IN#14=1

Remarks:

If the job name to be loaded already exists, the existing job will be deleted and the new job will replace it.

6.10.3.2 SAVEJ

Function:

The SAVEJ (save job) command specifies a single job or related jobs to be saved from the ERC Controller to the remote computer.

Format:

For direct specification of the job name.

SAVEJ JOB:job name job type extension (IF syntax)

For specification of job names through the pattern input of the input group.

SAVEJ IG#input group number job type extension (IF syntax)

Example:

For direct specification of the job name.

SAVEJ JOB:123 JBR IF IN#14=1

For specification of job names through the pattern input of the input group.

SAVEJ IG#01 JBI IF IN#14=1

Remarks:

None

6.10.3.3 DELETEJ

Function:

The DELETEJ (delete job) command deletes all jobs in the ERC Controller are deleted except for the current active job (which contains the delete instruction). In Software Versions 4.XX and higher the DELETEJ command can specify that a single job or related jobs only to be deleted. The DELETEJ command does not allow for job specification through the pattern input of the input group.

Format:

For deletion of all jobs except the current acive job.

DELETEJ (IF syntax)

For direct specification of the job name. Supported in Version 4.XX software and higher only.

DELETEJ JOB:job name job type extension (IF syntax)

Example:

For deletion of all jobs except the current acive job.

DELETEJ IF IN#14=1

For direct specification of the job name. Supported in Version 4.XX software and higher only.

DELETEJ JOB:123 JBR IF IN#14=1

Remarks:

If a job with delete protection is encountered, that job will not be deleted and the ERC Controller will issue a 1570 alarm. Software Version 4.21 and higher will generate a 1570 alarm if the job to be deleted does not exist. No communication occurs between the ERC Controller and the host computer with this command.

6.10.4 Load/Save Function for Variables

6.10.4.1 Introduction

Variable data can be loaded to the ERC Controller from a remote computer, and saved from the ERC Controller to a remote computer. Position data from a vision system or a decision result obtained by a remote computer can be loaded to the ERC Controller as variable data. The use of position variable data along with a position correction function (e.g. parallel shift) can be used to modify a programs path to correspond to a workpiece's actual position. A decision result contained within variable data can be used in conjunction with a conditional jump instruction to determine appropriate jobs or actions.

6.10.4.2 Variable Types

The following types of variables are available within the ERC Controller: Byte, Integer, Double Precision, Real, Position, and External Axis Position. A description of each is provided below.

- BYTE
  100 variables designated by B00 to B99.
  Values range from 0 to 255

- INTEGER
  100 variables designated by I00 to I99.
  Values range from -32768 to +32767

- DOUBLE PRECISION
  100 variables designated by D00 to D99.
  Values range from -2147483648 to +2147483647

- REAL
  100 variables designated by R00 to R99.
  Values range from -1.70141E+38 to +1.70141E+38
  Seven (7) significant digits

- POSITION (Robot Axes)

  64 variables designated by P00 to P63.
  Stores manipulator and wrist position data for up to 6 axes.
  Values stored in pulse counts or rectangular coordinates position data.
  Pulse Counts: S, L, U, R, B, T(Units of pulse counts)
  Rectangular Coordinates: X, Y, Z, (Units of mm)
  (3 significant decimal places)
  TX, TY, TZ (Units of degrees)
  (3 significant decimal places)

• EXTERNAL AXIS POSITION

64 variables designated by EX00 to EX63 .
Stores position data for up to 6 external axes.
Values stored in pulse counts or rectangular coordinates position data (only for 7, 8, and 9 when used for linear track axes).

Pulse Counts: 7, 8, 9, 10, 11, 12	(Units of pulse counts)
Rectangular Coordinates: X, Y, Z	(Units of mm) (3 significant decimal places)
10, 11, 12	(Units of pulse counts)

6.10.4.3 Command Description

A description of the Load and Save commands is provided in the following two sections. The description details the command name, function, format and an example. Required command components are depicted in capital letters. "Variable number" would be replaced with the number of the desired variable. Refer to the "K" SERIES ROBOT OPERATOR'S MANUAL for a description of the INFORM programming language.

6.10.4.3.1 LOADV

Function:

The LOADV (load variable) command specifies a variable number to be loaded from the remote computer to the ERC Controller.

Format:

LOADV B(Byte-type variable number)
LOADV I(Integer-type variable number)
LOADV D(Double Precision-type variable number)
LOADV R(Real-type variable number)
LOADV P(Position-type variable number)
LOADV EX(External Axis Position-type variable number)

Examples:

LOADV B01
LOADV P07
LOADV R17

Remarks:

None

Example 6.5 *LOADV:*

The following transactions are used to transmit the value contained in a byte variable from the remote computer to the ERC Controller.

1. The ERC transmits an ENQ to establish the data link with the remote computer.
2. The remote computer responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the ERC transmits the command sequence to request the upload of a byte variable. Header code 03,051 instructs the remote computer to send a byte variable.
4. The remote computer responds with an ACK1.
5. The ERC issues an EOT. The ERC then waits for the variable data.
6. The remote computer transmits an ENQ to establish a data link with the ERC.
7. The ERC responds to the ENQ with an ACK0.
8. After receipt of the ACK0, the remote computer transmits the command sequence to load a byte variable. Header code 03,001 instructs the ERC to load a byte variable. The value of the byte variable is contained within the text field.
9. The ERC responds with an ACK1.
10. The remote computer issues an EOT to signal the end of communications.

Figure 6.15 depicts a transmission sequence, initiated by the ERC, to load a byte variable from a remote computer to the ERC Controller.

LOADV
ERC <--- REMOTE COMPUTER

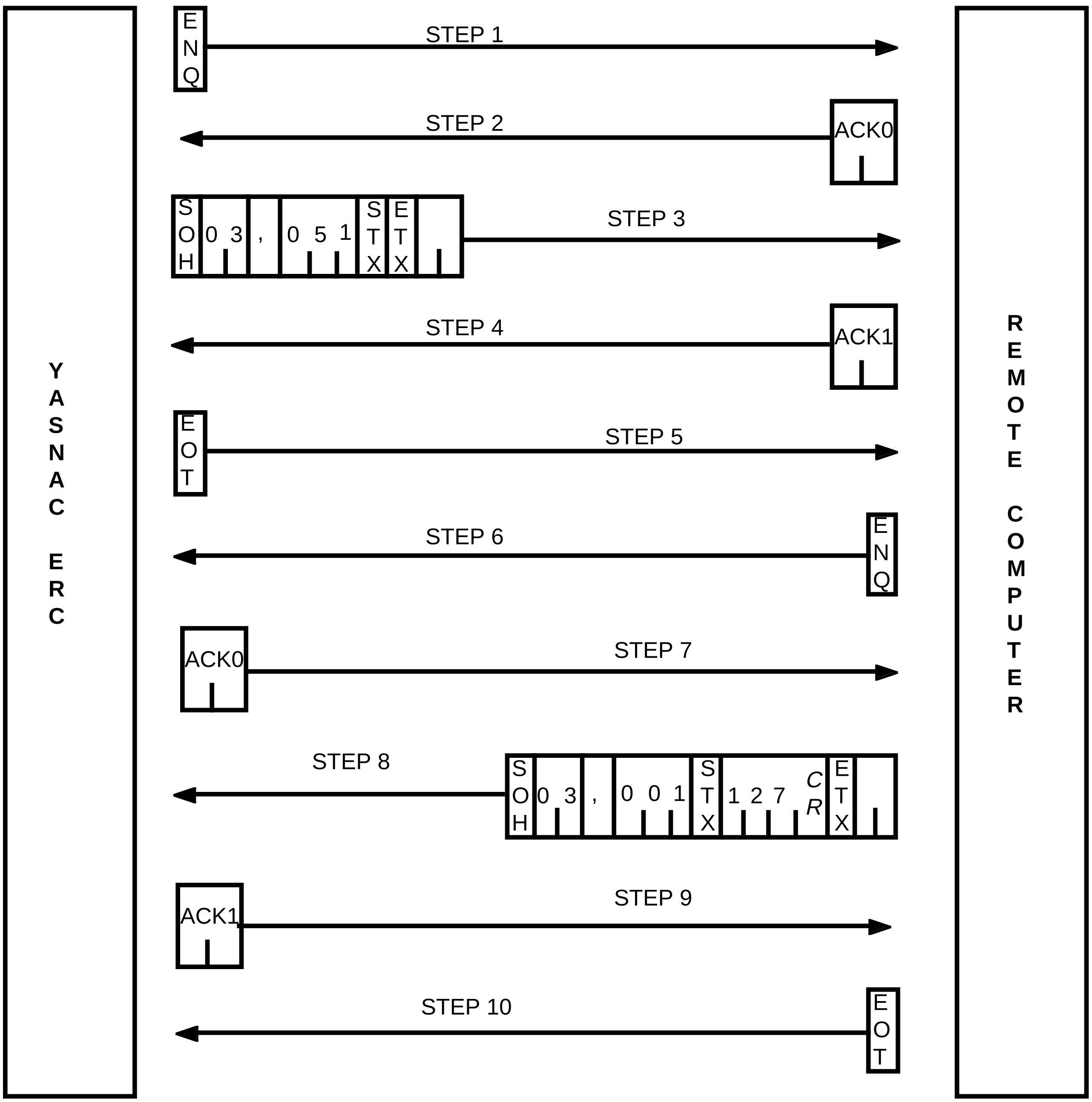


Figure 6.15 LOADV Transmission Sequence

6.10.4.3.2 SAVEV

Function:

The SAVEV (save variable) command specifies a variable number to be saved from the ERC Controller to the remote computer.

Format:

SAVEV B(Byte-type variable number)
SAVEV I(Integer-type variable number)
SAVEV D(Double Precision-type variable number)
SAVEV R(Real-type variable number)
SAVEV P(Position-type variable number)
SAVEV EX(External Axis Position-type variable number)

Examples:

SAVEV B01
SAVEV P07
SAVEV R17

Remarks:

None

Example 6.6 *SAVEV*

The following transactions are used to transmit the value contained in a real variable from the ERC Controller to the remote computer.

1. The ERC transmits an ENQ to establish the data link with the remote computer.
2. The remote computer responds to the ENQ with an ACK0.
3. After receipt of the ACK0, the ERC transmits the command sequence to save a byte variable. Header code 03,004 instructs the remote computer to save a real variable. The value of the real variable is contained within the text field. Each digit is treated as an individual character (including the decimal point).
4. The remote computer responds with an ACK1.
5. The ERC issues an EOT. The ERC is in a waiting condition for a response from the remote computer that the variable save has been successful.
6. The remote computer transmits an ENQ to establish a data link with the ERC.
7. The ERC responds to the ENQ with an ACK0.
8. After receipt of the ACK0, the remote computer transmits the command sequence indicating a successful save. Header code 90,000 indicates to the ERC, a response from a remote computer. The response code is contained within the text field.
9. The ERC responds with an ACK1.
10. The remote computer issues an EOT to signal the end of communications.

Figure 6.16 on the following page depicts a transmission sequence, initiated by the ERC, to save a real variable from the ERC Controller to a remote computer.

SAVEV
ERC ---> REMOTE COMPUTER

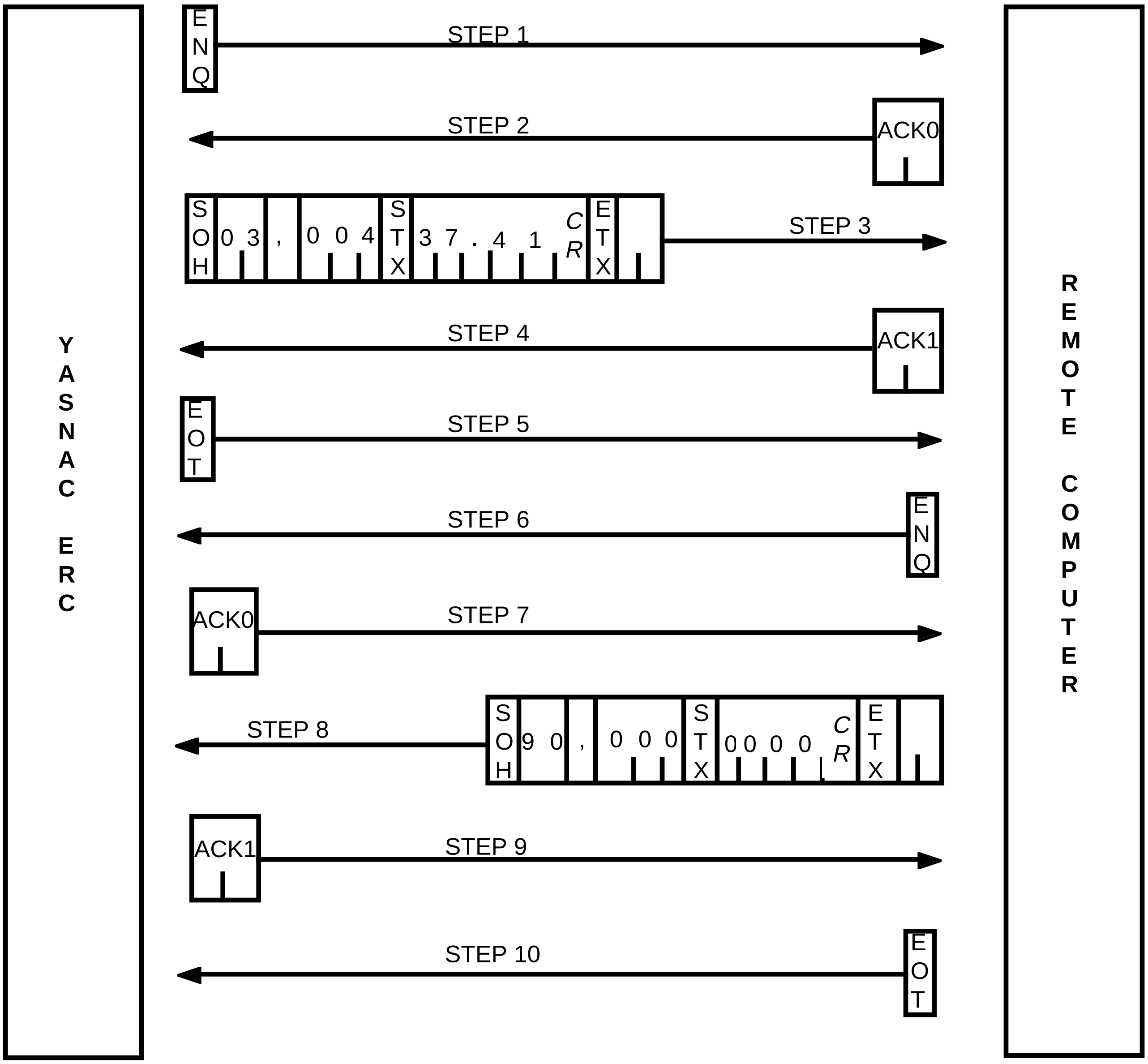


Figure 6.16 SAVEV Transmission Sequence

6.11 COMMAND TRANSACTION HEADER LIST

Table 6-10 provides a breakdown of the command transaction header number, sub code number and a description of the resulting command. Only those numbers currently assigned are displayed.

Table 6-10 Command Transaction Codes

Header No.	Sub Code No.	Description
01	000	Command from a remote computer.
02	000	
	001	"job name".JBI
		Indicates independent job data.
	002	"job name".JBR
		Indicates related (master) job data.
	011	WEAV.DAT
		Indicates weave data.
	012	TOOL.DAT
		Indicates tool data.
	013	UFRAME.DAT
		Indicates user coordinate data.
	015	CV.DAT
		Indicates conveyer condition data.
	016	SENSOR.DAT
		Indicates locus correction condition data.
	017	COMARC2.DAT
		Indicates COM-ARC 2 condition data.
	018	PC1PC2.DAT
		Indicates phase comprehension data.
	030	SYSTEM.DAT
		Indicates system data.

Table 6-10 Command Transaction Codes Continued

Header No.	Sub Code No.	Description
02	051	"job name".JBI Request for independent job data.
	052	"job name".JBR Request for related (master) job data.
	061	WEAV.DAT Request for weave data.
	062	TOOL.DAT Request for tool data.
	063	UFRAME.DAT Request for user coordinate data.
	065	CV.DAT Request for conveyer condition data.
	066	SENSOR.DAT Request for locus correction condition data.
	067	COMARC2.DAT Request for COM-ARC 2 condition data.
	068	PC1PC2.DAT Request for phase comprehension data.
	080	SYSTEM.DAT Request for system data.
03	000	
	001	Byte-type variable
	002	Integer-type variable
	003	Double precision-type variable
	004	Real-type variable
	005	Position-type variable (pulse data)
	006	Position-type variable (rectangular data)
	007	External Axis-type variable (pulse data)
	008	External Axis-type variable (rectangular data)
	050	
	051	Request for Byte-type variable
	052	Request for Integer-type variable
	053	Request for Double Precision-type variable
	054	Request for Real-type variable
	055	Request for Position-type variable (pulse data)
	056	Request for Position-type variable (rectangular data)
	057	Request for External Axis-type variable (pulse data)
	058	Request for External Axis-type variable (rectangular data)

Table 6-10 Command Transaction Codes Continued

Header No.	Sub Code No.	Description
90	000	0000 or "error code" (4 digits) Response to a transmission following execution of a command. The text field will contain either 0000 or an error code. 0000 indicates transaction was OK. Any 4 digits other than 0000 indicates an error condition. Table 4-11 provides a list of thepossible error codes.
90	001	"data" (variable number of digits/data) Response to a transmission following execution of a command. Contains data in the text field. The amount and type of data depend upon the command which was sent. All data elements are sent as individual ASCII characters. For example, if the two real values 47.2 and 7345.97 are being passed they will be sent as 47.2,7345.97CR. Each digit, including the decimal points and commas, are sent as individual ASCII characters. The carriage return (CR) is a single ASCII character as well. There are no blank spaces between data items. All data items are separated by commas.

6.12 INTERPRETER MESSAGE LIST

After receiving a command from a remote computer, the ERC Controller will issue an error message if the command cannot be executed. The error message, called the interpreter code, will be transmitted to the remote computer in the text field of a command with a 90,000 header/sub code number. Table 6-11 provides the interpreter message code and error message.

The interpreter message codes are broken down into the following five categories:

1xxx: Command test

2xxx: Command execution mode error

3xxx: Command execution error

4xxx: Job registration error

5xxx: File text error

Table 6-11 Interpreter Code and Error Message

Code No.	Error message
1010	Command failure
1011	Command operand number failure
1012	Command operand value excessive
1013	Command operand length failure
2010	During robot operation
2020	During T-PENDANT
2030	During panel HOLD
2040	During external HOLD
2050	During command HOLD
2060	During error alarm
2070	In Servo OFF or stopping by a panel HOLD
3010	Servo power on
3040	Set home position
3070	Current position is not input
3080	END command of job (except master job)

4010	Shortage of memory capacity (job registration)
4012	Shortage of memory capacity (position data registration)
4020	Job edit prohibit
4030	Job of same name exists
4040	No desired job
4060	Set execution
4120	Position data broken
4130	No position data
4150	END command of job (except master job)
4170	Instruction data broken
4190	Unsuitable characters in job name exist
4200	Unsuitable characters in job name exist
4230	Instructions which cannot be used by this system exist
5110	Instruction syntax error
5120	Position data fault
5130	Neither NOP or END exists
5170	Format error
5180	Data number is inadequate
5200	Data range exceeded

6.13 TRANSMISSION ALARMS

When the ERC detects a transmission error an alarm code error number will display on the ERC operator's panel CRT. If the ERC is in remote mode when an alarm occurs, it will change to stand-alone mode. Due to the transmission problem, the alarm code cannot be transmitted to the remote computer.

The ERC Controller can be reset by pressing the RESET soft key. Standalone mode is now operational. Remote mode can be entered and communications re-establish with the remote computer by pressing the REMOTE key.

The transmission alarm codes are broken down into the following categories:

Data Communications with DCI Alarms

1550: Load instruction execution error

1560: Save instruction execution error

1570: Delete instruction execution error

Data Communications Alarms

1700: Transmission error

1710: Receiving error

1720: Hardware error

1730: System block

Alarm codes 1550, 1560 and 1570 display the alarm code and a three digit data field indicating the cause of the alarm. Each of the instruction execution error alarms (Load, Save, Delete) use the same three digit data code to indicate the cause of the alarm. Table 6-12 provides the Data Communications with DCI alarm data codes and cause. As the three digit data fields and corresponding cause are the same for each of the instruction executions errors (Load, Save, Delete), they are only described once.

Table 6-12 Data Communications with DCI Data Codes and Cause for Alarms 1550, 1560 and 1570

Data Code No.	Alarm Code Cause
001	Insufficient memory
002	Job protected from edit
003	Job name duplication
004	Job does not exist
012	Faulty position data
013	Undefined position variable
017	Faulty instruction
019	Job name contains illegal character
020	Label contains illegal character
023	Unacceptable character
024	Syntax error
104	File does not exist in the remote computer
111	Syntax error
112	Incorrect position data
113	Job does not contain a NOP or END command
117	Format error
118	Incorrect number of data
120	Data range surpassed
122	Faulty file
124	Command cannot be executed in remote mode
211	System block error (EOT while waiting for ACK)
212	System block error (EOT while waiting for ENQ)
213	System block error (EOT before receiving the last block)
214	System block error (Code other than EOT after receiving the last block)

215	System block error (RVI)
221	Transmitting error (excessive NAK retry)
222	Transmitting error (Timer A time-out after retry)
223	Transmitting error (ACK0/ACK1 incorrect order after retry)
231	Receiving error (Timer A time-out while waiting for ACK after ENQ) (Timer A time-out waiting for ENQ)
232	Receiving error (Timer A time-out while waiting for text)
233	Receiving error (short heading length)
234	Receiving error (long heading length)
235	Receiving error (incorrect heading number)
236	Receiving error (text length exceeds 256 characters)
240	Software error
241	Hardware error (overrun error)
242	Hardware error (parity error)
243	Hardware error (framing error)
244	Hardware error (Timer A time-out)
245	Hardware error (Timer B time-out)

Table 6-13 Data Communications Alarm Codes and Cause

Code No.	**Alarm Code Cause**
1700	(1) Sequence retry excessive (2) Message retry excessive (3) Alternate acknowledge error

NOTE: For 1700 the ERC transmits an EOT and releases the data link.

Code No.	**Alarm Code Cause**
1710	(1) Receiving timer out (Timer A) (2) Receiving timer out (Timer B) (3) Short heading length (4) Long heading length (5) Heading number error (6) Text length is over 256 characters

NOTE: For 1710 cause 3-6 the ERC transmits an EOT and releases the data link.

Code No.	**Alarm Code Cause**
1720	(1) Overrun error (2) Parity error (3) Framing error (4) Receiving station hardware error (5) Transmitting station time-out error
1730	(1) EOT while waiting for ACK (2) EOT while waiting for ENQ (3) EOT before receiving the last block (4) Code other than EOT after receiving the last block (5) RVI

NOTE: For 1730 the ERC transmits an EOT and releases the data link.

6.14 JOB DATA FORMAT

The format of a job data file is provided below. Key words (commands) are identified in capital letters. Variable data are identified with small letters.

/DIR master job name : Directory of related jobs contained in master job

/JOB : Identifies the file as job data

//NAME job name : Identifies the job name

//POS : Identifies the following data as position data

///NPOS i, j, k, l : Defines the number of positions

i = Number of taught positions (robot axes)

j = Number of taught positions (external axes)

k = Number of variable positions (robot axes)

l = Number of variable positions (external axes)

///TOOL n : Tool number used during TEACH mode (1 - 7)

///PULSE : Indicates position data (pulse counts)

///RECTAN : Indicates position data (rectangular coordinates)

///RCONF l, m, n,o,p : Robot configuration data

l = 0 wrist flip, 1 no wrist flip

m = 0 elbow up, 1 elbow down

n = 0 upper arm front, 1 upper arm back

o = 0 R<180°, 1 R>180°

p = 0 T<180°, 1 T>180°

//INST : Identifies the instructions section (INFORM programming language instructions)

///DATE y/m/d h:m : Identifies the date

y = year

m = month

d = day

h = hour

m = minute

///ATTR a-p	: Job data attributes
	a-g= Reserved for future use
	h = Multilayer Weld reverse job, 0 normal, 1 reverse job
	i = Job fault, 0 normal, 1 fault
	j = Home position of PT coordinates, 0 normal, 1 home
	k = Ref. position of 3D shift, 0 normal, 1 ref. position
	l = Reserved for future use
	m = Position data format, 0 pulse, 1 rectangular
	n = Data saved, 0 not saved, 1 saved
	o = Reserved for future use
	p = Edit prohibit, 0 normal, 1 prohibit job edit
///FRAME coordinate system	: Identifies the job coordinate system
	BASE = base coordinates
	ROBOT = robot coordinates
	N = user coordinate number (1 - 8)

6.14.1 Independent Job Data Example

```
/JOB
//NAME EXAMPLE
//POS
///NPOS 3,0,0,0
///TOOL 0
///PULSE
C000=2257,30448,-241,-7641,113,-43
C001=-71,8864,66106,66765,50543,-32254
C002=-885,-65098,6543,8776,986,10223
//INST
///DATE 1990/10/06 9:00
///ATTR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
///FRAME BASE
NOP
MOVC C000
MOVC C001
MOVC C002
RET
END
```

6.14.2 Master Job Data Example

```
/DIR MASTER1
JOB1, TEST002, TOPJOB,TEMP,
.
.
.
DEBUR1, DBURX
/JOB
//NAME MASTER1
//POS
///NPOS 3,0,0,0
///TOOL 0
///PULSE
//INST
```

```
///DATE 1990/10/15 14:00
///ATTR 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
///FRAME BASE
NOP
.
.
.
CALL JOB:JOB1
.
.
.
CALL JOB:TEST002
.
.
etc.
.
.
END
```

7.0 ERC COMMUNICATIONS WITH A PERSONAL COMPUTER

7.1 INTRODUCTION;

The connection of a personal computer to the ERC Controller provides a variety of functionality, as described in Section 6.1. The PC may simply function as an external storage device for the ERC, providing the ability to load and save data files. User-developed application programs may be implemented on the PC to control the robot through the use of the command functions and protocol defined within Section 6.

This portion of the document contains information regarding the communications between the ERC Controller and an IBM PC, XT or AT personal computer. As brands of personal computers may vary in their compatibility with the IBM PC standards, the information presented within this section may not reflect the requirements of the actual PC used. Refer to the reference manuals accompanying the personal computer being used.

7.2 REQUIREMENTS;

The requirements for the personal computer are minimum. The PC must be able to interface with the ERC Controller through a serial communications link as described in Section 6.2.2. Additionally, the PC must be able to send and receive data, and run application software which is compatible with the protocols described in Section 6. Beyond that, the requirements for the PC are entirely dependent upon the user's application programs, operating system, storage requirements, etc.

7.2.1 Communications Cable

A personal computer can use one of two devices for the transfer of asynchronous data. A serial/parallel adapter card is available for IBM PC AT compatible computers. An asynchronous communications adapter is available for IBM PC and XT compatible computers.

7.2.1.1 IBM PC AT Compatible

A communications cable connects the ERC Controller's RS-232C port to the IBM PC AT compatible's serial port connector on the serial/parallel adapter card. The standard serial port for the IBM PC AT is a DB-9 male connector. Thus, a communications cable with a female DB-9 connector is required for the PC AT.

The ERC connection end of the communications cable is described in Section 6.2.2.

Figure 7.1, on the following page, illustrates the IBM PC AT DB-9 serial port.

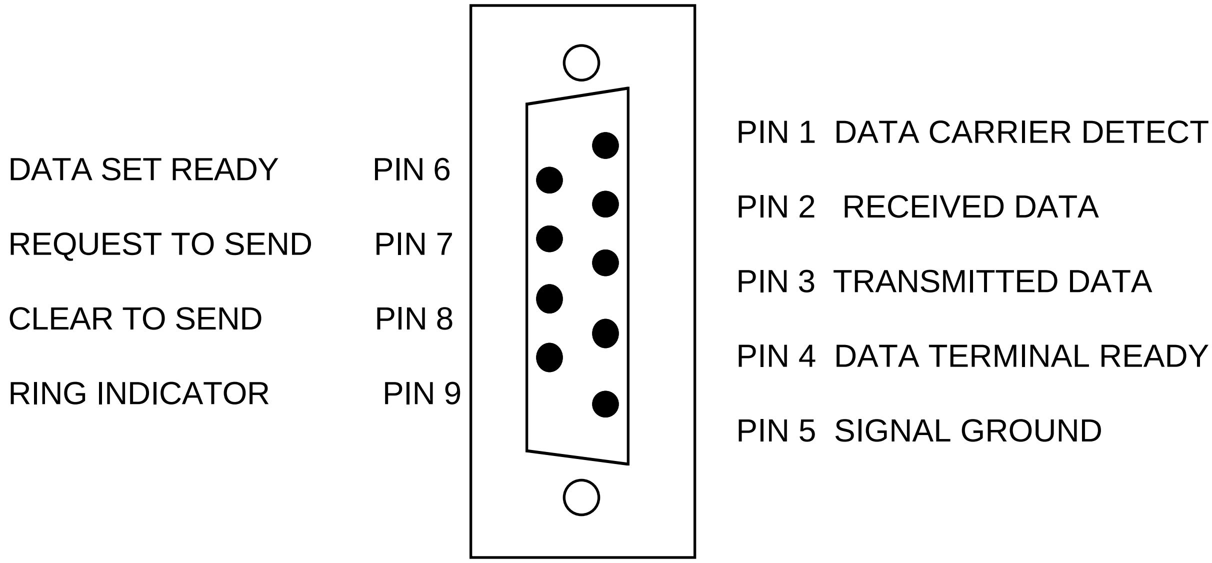


Figure 7.1 IBM PC AT DB-9 serial port

Table 7-1 IBM PC AT DB-9 pin description

Pin Number	Description
1	CD, Carrier Detect
2	RD, Received Data
3	TD, Transmitted Data
4	DTR, Data Terminal Ready
5	SG, Signal Ground
6	DSR, Data Set Ready
7	RTS, Request To Send
8	CTS, Clear To Send
9	RI, Ring Indicator

Normally, PCs interface through the serial port to a communications device. The communications device interfaces with another communications device which, in turn, interfaces with another computer. This configuration allows two computing devices to communicate. However, in this instance, the PC will be communicating directly with another computer, the ERC Controller, without communication devices. The communications cable required to accomplish this interface resembles a null modem.

A null modem is a method of connecting two computing devices, such that each device thinks that it is interfaced to a communications device. Figure 7.2 illustrates the communications cable required for the connection of the ERC's DB-25 connector to the PC AT's DB-9 connector.

➪ *CAUTION!*

The RS-232C standard limits the length of the cable to 50 feet. A longer cable may be used, but should be thoroughly tested in the working environment to ensure signal quality.

➪ *CAUTION!*

To ensure signal quality, always use a shielded cable.

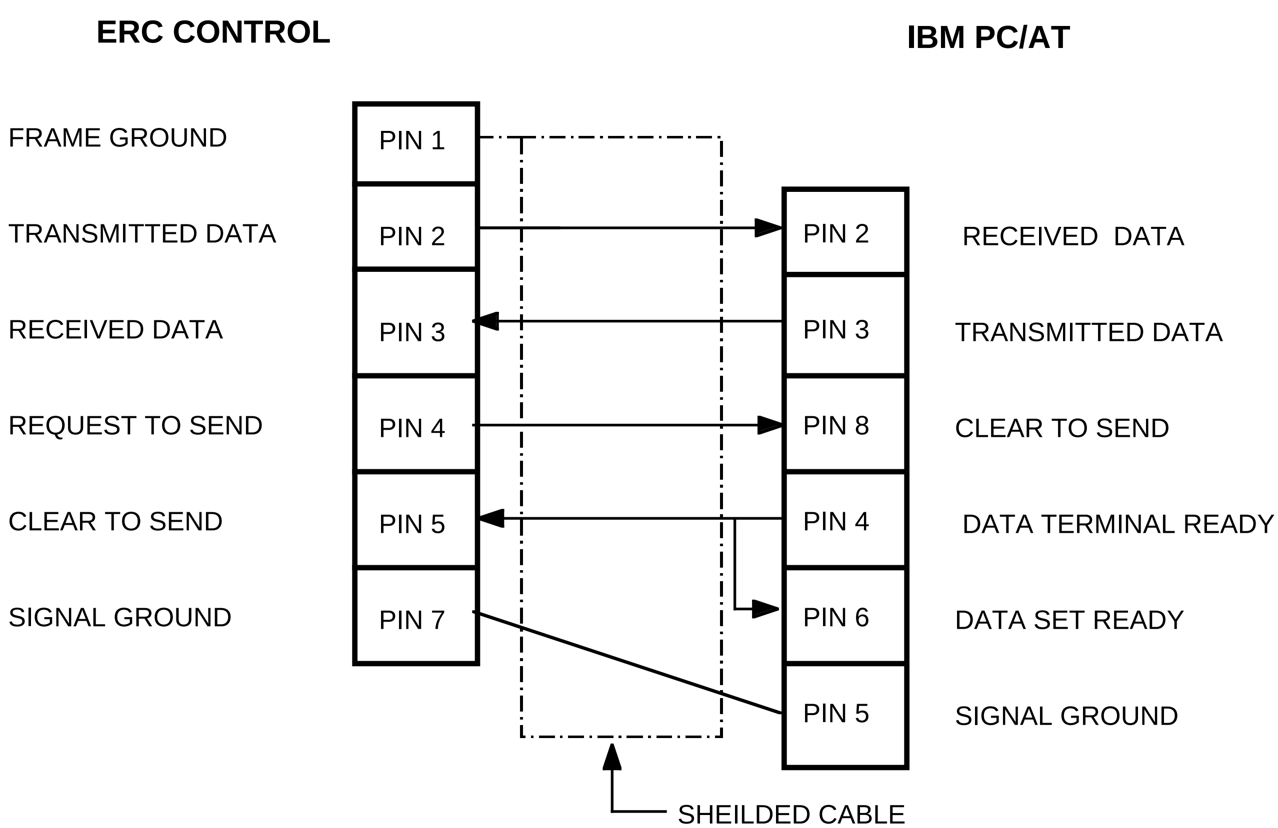


Figure 7.2 ERC to IBM PC AT Communication Cable

7.2.1.2 IBM PC or XT Compatible

A communications cable connects the ERC Controller's RS-232C port to the IBM PC or XT compatible's connector on the asynchronous communications adapter card. The standard serial port for the asynchronous communications adapter is a DB-25 male connector. Thus, a communications cable with a female DB-25 connector is required for the PC or XT. The ERC connection end of the communications cable is described in Section 6.2.2.

Figure 7.3, below, illustrates the IBM PC or XT asynchronous communications adapter DB-25 serial port.

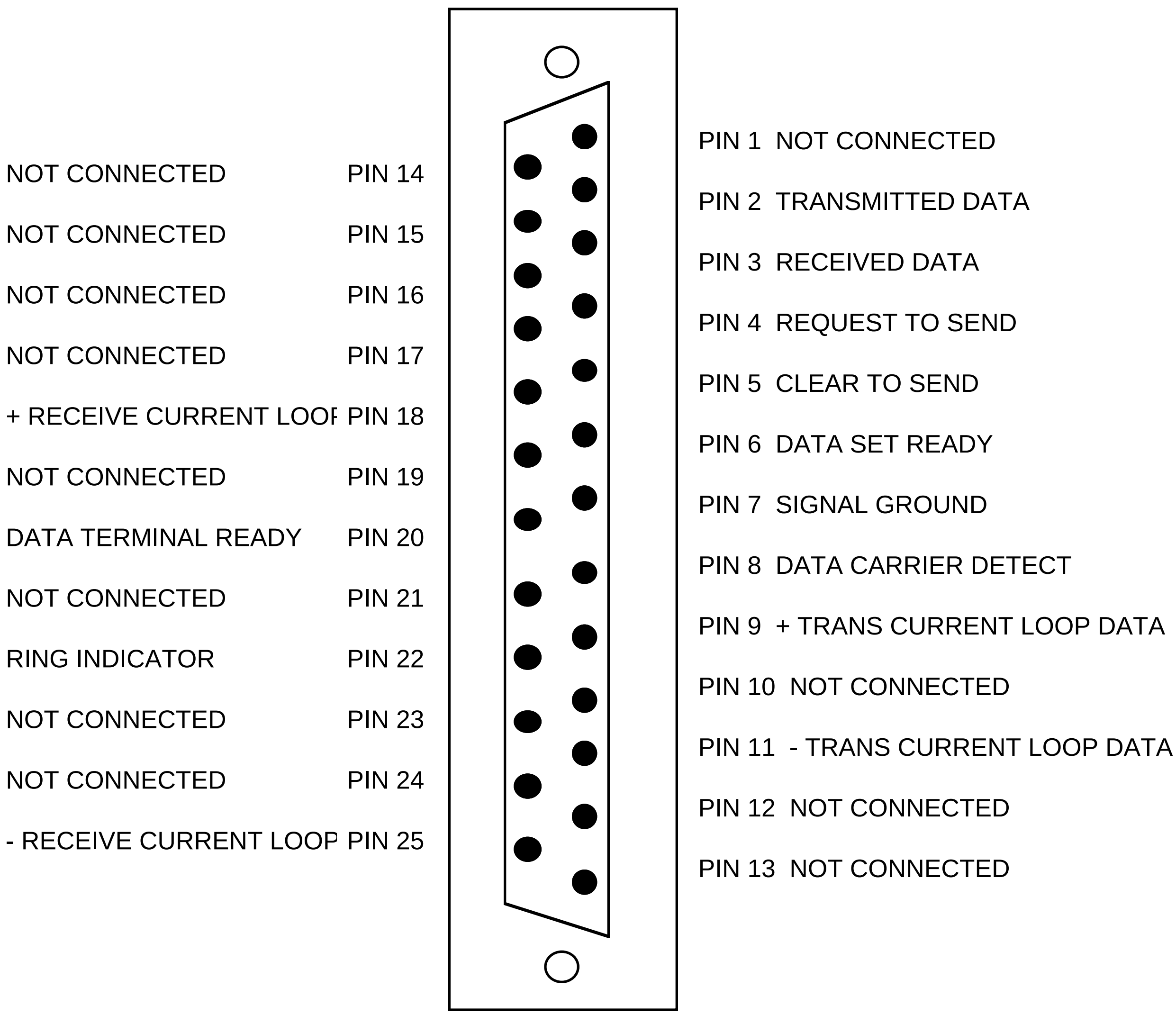


Figure 7.3 IBM PC or XT DB-25 Serial Port

Table 7-2 provides a description of the IBM PC or XT asynchronous communications adapter DB-25 serial port.

Table 7-2 IBM PC or XT DB-25 Pin Description

Pin Number	Description
1	Not Connected
2	TD, Transmitted Data
3	RD, Received Data
4	RTS, Request To Send DTR, Data Terminal Ready
5	CTS, Clear To Send
6	DSR, Data Set Ready
7	SG, Signal Ground
8	DCD, Data Carrier Detect (Received Line Signal Detector)
9	+ Transmit Current Loop Data
10	Not Connected
11	- Transmit Current Loop Data
12	Not Connected
13	Not Connected
14	Not Connected
15	Not Connected
16	Not Connected
17	Not Connected
18	+ Receive Current Loop Data
19	Not Connected
20	DTR, Data Terminal Ready
21	Not Connected
22	RI, Ring Indicator
23	Not Connected
24	Not Connected
25	- Receive Current Loop Return

Normally, PCs interface through the serial port to a communications device. The communications device interfaces with another communications device which, in turn, interfaces with another computer. This configuration allows two computing devices to communicate. However, in this instance, the PC will be communicating directly with another computer, the ERC Controller, without communication devices. The communications cable required to accomplish the interface resembles a null modem.

A null modem is a method of connecting two computing devices, such that each device thinks that it is interfaced to a communications device. Figure 7.4 illustrates the communications cable required for the connection of the ERC's DB-25 connector to the PC's or XT's asynchronous communications adapter DB-25 connector.

➪ *CAUTION!*

The RS-232C standard limits the length of the cable to 50 feet. A longer cable may be used, but should be thoroughly tested in the working environment to ensure signal quality.

➪ *CAUTION!*

To ensure signal quality, always use a shielded cable.

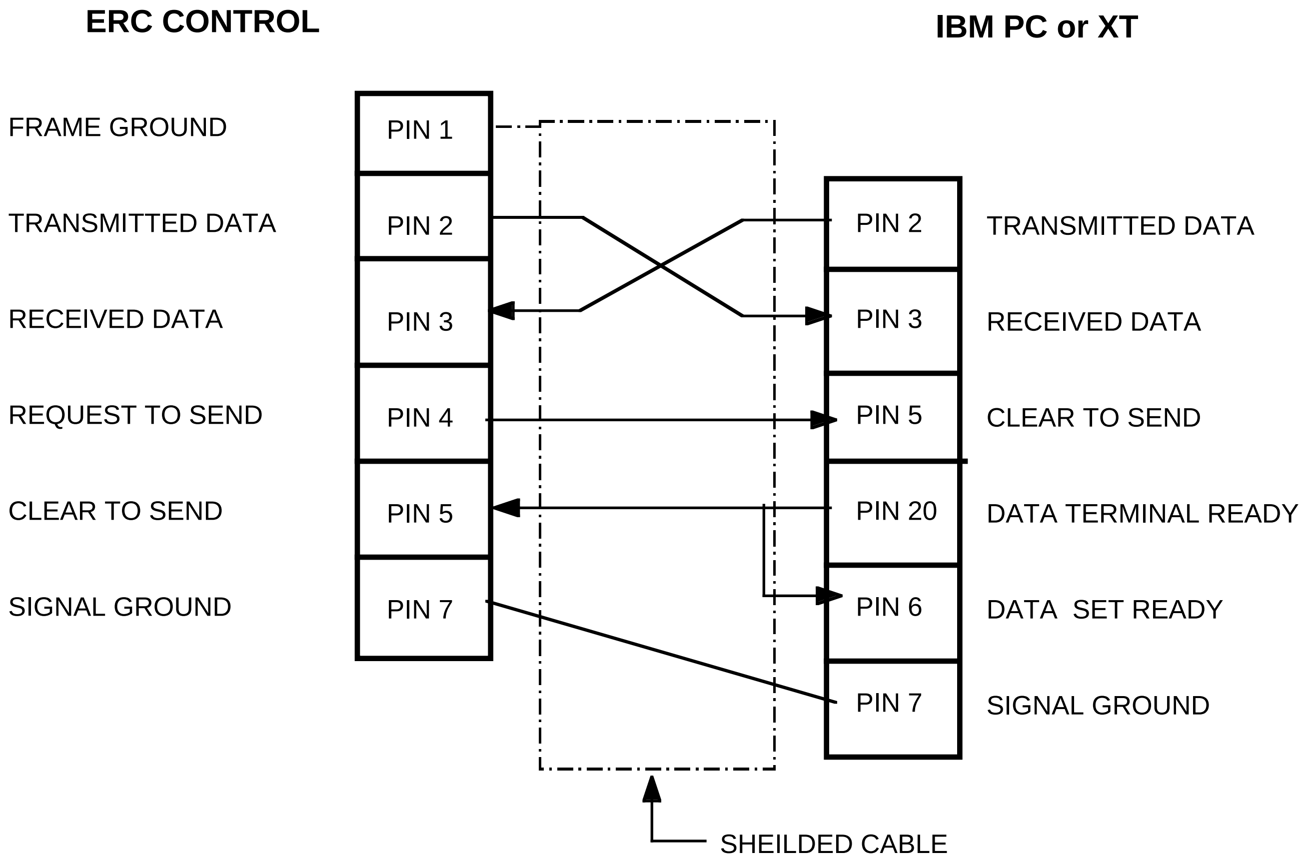


Figure 7.4 ERC to IBM PC or XT Communication Cable.

7.2.2 Communication Parameters

The personal computer's serial port should be set such that the communication parameters match those of the ERC Controller, described in Section 6.3. The method of setting these parameters will depend upon the serial communications board and the operating system of the PC.

If the personal computer is running under DOS, the MODE command can be used to set the parameters for the serial port. The following example of the MODE command sets the parameters of serial port COM1 to 9600 bps, 8 data bits, even parity, and 1 stop bit.

MODE COM1 9600, E, 8, 1

The command format of the MODE command may depend upon the version of DOS used. Refer to a DOS user's guide for a detailed explanation of the MODE command specific to the version of DOS utilized.

7.2.3 Application Program

Once communications between the ERC Controller and the PC have been established, the customer can create an application program, utilizing the commands and protocol format provided within Section 6.0, to transmit data and control the robot from the PC. Normally, a high-level programming language such as BASIC, PASCAL, or C would be utilized to create the application program.

There is an application program, entitled MY-BASIC, available for the IBM PC AT, which provides a set of call-able functions through BASICA. The MY-BASIC application program and requirements are described in the MY-BASIC OPERATIONS MANUAL.

APPENDIX A   GLOSSARY

ACK (Acknowledge): A control character used to verify proper communication between a transmitting and receiving station.

Adapter: Any printed circuit board that plugs into one of the slots on a PC's motherboard.

ANSI (American National Standards Institute): The coordinating organization for voluntary standards groups within the United States.

ASCII (American Standard Code For Information Interchange): An eight (8) level code, adopted by the American National Standards Institute, for data transmission.

Asynchronous: Having a variable or random time interval between successive characters, operations, or events. Transmission in which each character, word, or small block, is individually synchronized (timed), usually by the use of start and stop bits.

AUTOEXEC.BAT: An optional file on the computer's boot disk, containing various instructions that the user wishes to have performed automatically when the system is turned on.

Baud: A unit of signaling speed. The speed in baud is the number of discrete conditions or signal events per second. (This is applied only to the actual signals on a communications line.) If each signal event represents only one bit condition, baud is the same as bits per second.

BASIC (Beginners All-Purpose Symbolic Instruction Code): A general-purpose programming language, designed to be easy to learn, remember, and use.

Batch File: Any file with an extension of BAT that is used in a manner similar to AUTOEXEC.BAT. The difference is that a file with a name other than AUTOEXEC.BAT is not executed automatically. Instead, the user must type the name of the file and then press the Enter key to execute the batch file.

BCC (Block Check Character): A number of characters added to the end of a block of data to allow the receiving device to check for transmission errors.

Binary: Pertaining to the base 2 system of numerical notation in which only the digits 0 and 1 are used.

Binary Synchronous Protocol (Bisync, BSC): A half-duplex, character-oriented, synchronous data communications protocol. Required line control codes are used according to specified rules to maintain communications.

Bit (Binary Digit): The smallest unit of information in a binary system of notation, represented by a 0 or 1.

Bit Oriented: A communications protocol where control information is encoded in one or more bits. As opposed to characters for control.

Block: Some set of contiguous bits and/or bytes which make up a quantity of information.

Boot: When a PC is turned on, a small program built into the system hardware is used to load a program from the DOS diskette. To boot the system means to start this process.

Boot Disk: The diskette that is used when the PC is turned on.

BPS (Bits Per Second): A measure of the rate at which bits are transmitted over a communications path every second. The BPS is not necessarily equal to the data signaling rate (Baud), which measures the rate of signal changes being made.

Byte: A series of eight (8) adjacent bits treated as a single unit of information.

Character: A letter, figure, number, punctuation, or other sign contained in a message. Additionally, there may be characters for special symbols and control functions.

Character Oriented: A communications protocol that carries control information encoded in one or more bytes. Also known as byte oriented.

Checksum: The summation of a group of data, which is provided within the group. The checksum is used for error checking purposes.

Code: A system of symbols and rules used in representing information.

Connector: A physical interface, such as EIA RS-232C, which typically contains a male and female component.

Control Character: A character whose occurrence in a particular context initiates, modifies, or stops a control function.

Conventional Memory: Random access memory installed between absolute addresses 0 and 640K.

CR (Carriage Return): A control character which advances the active print or display position to the first column on the same line.

CTS (Clear To Send): A control line between a modem and a controller used to operate over a communications line.

Data: A representation of facts, concepts, or instructions in a formalized manner suitable for communication.

Data Communication: The function of transporting encoded information from one point to another.

DCE (Data Communications Equipment): The equipment installed at the user's premises which provides all the functions required to establish, maintain and terminate a connection, the signal conversion, and control between the data terminal equipment and the common carrier's line.

Decimal: Pertaining to the base 10 system of numerical notation, in which the digits 0 through 9 are used.

Diskette: A flexible magnetic disk, permanently enclosed in a protective PVC jacket. Often called a floppy disk.

DLE (Data Link Escape): A control character used to modify the meaning of a limited number of subsequent characters.

DOS (Disk Operating System): The set of programs that are required to support the basic operations of the computer system.

DTE (Data Terminal Equipment): The equipment comprising the data source or data termination points.

DTR (Data Terminal Ready): An interface control signal from a DTE, indicating to DCE that the terminal is ready for transmission.

Duplex (Full Duplex): A facility which permits transmission in both directions simultaneously.

EBCDIC (Extended Binary Coded Decimal Interchange Code): An eight (8) bit code, standard for many IBM systems, which provides 256 possible character combinations.

EIA (Electronic Industries Association): The U.S. national organization of electronic manufacturers responsible for the development and maintenance of industry standards for the interface between data processing and data communications equipment.

ENQ (Enquiry): A control character used to request a response from a receiving device.

EOT (End Of Transmission): A control character signifying the end of all transmitted data corresponding to a message.

Error: A general term referring to any deviation from the correct or true value.

ETB (End Of Transmission Block): A control character signifying the end of a particular block of transmitted data.

ETX (End Of Text): A control character used to signal the receiving station that all information data has been sent.

Format: The surface of a diskette must be magnetically prepared before data can be written to it.

Forms Effectors: Control characters intended for control of information on printing and/or display devices.

File: A collection of data records or organized information.

Flow Control: A technique necessary to control the flow of data.

Format: A structure of a message or data such that specific controls or data can be identified by its position during processing.

Ground: Electrical common conductor.

Half Duplex: A circuit designed for transmission in either direction, but not both directions simultaneously.

Hardware: Physical equipment, as opposed to a computer program.

Hexadecimal: Pertaining to the base 16 numbering system, in which the digits 0 through 9 and the letters A-F are used to express the hexadecimal equivalents of decimal numbers 0-15.

Host Computer: The controlling computer.

Information: Data that has intrinsic meaning to a user.

Interface: A common boundary or connection between computer systems or parts of a system. Usually interface describes the physical connection between two systems or two devices. However, interface can also describe the point of interconnection of two components, and how they must exchange signals according to some hardware or software protocol.

Job: A set of data, including programs, files and instructions for a computer.

K: When referring to data storage capacity, K equals two to the tenth power, or 1,024 in decimal notation.

LED (Light Emitting Diode): A semiconductor device that emits visible light from its surface.

LF (Line Feed): A control character which causes the active printing or display position to advance to the same column position on the next line.

M: When referring to data storage capacity, M equals two to the twentieth power, or 1,048,576 in decimal notation.

Mainframe Computer: A large computer capable of processing large amounts of data very fast. May contain several processing units working collectively.

Memory: That part of a computer in which data is stored for later retrieval. The contents of memory can be permanently fixed (ROM) or changeable as needed (RAM).

Micro Computer: A complete small computing system whose processing blocks are made of semiconductor integrated circuits. The functions are similar to a mini computer with the only difference being price, size, speed of execution, and computing power. Commonly known as a personal computer (PC).

Mini Computer:. A complete small computer.

Modem: A contraction of "modulation-demodulation". A conversion device installed in pairs at each end of an analog communications line. The modem at the transmitting end modulates digital signals received from a computer or terminal. The modem on the receiving end demodulates the incoming analog signals, converting them back into their original format and passes them to the destination station.

NAK (Negative Acknowledgment): A control character used to indicate improper communication between a transmitting and receiving device.

Null Modem: A device that connects two DTEs by emulating the physical connections of a DCE device. Also known as a null modem cable.

Packet: A group of bits treated as a unit for communication and routing purposes.

Parallel Interface: An interface where an entire group of bits is transmitted at one time by sending each bit over a separate wire.

Parallel Transmission: A method of information transfer in which bits composing a character are sent simultaneously.

Parity Bit: A non-information bit added to data for the purpose of error checking.

PC (Personal Computer): A common referral to a micro computer.

Port: The entrance or physical access point to a computer, device, or network where signals may be supplied, extracted, or observed.

Protocol: A set of rules and conventions that govern the orderly and meaningful exchange of information and error control between points.

RAM (Random Access Memory): The portion of memory in which the user may store data.

Read: To scan data stored on a disk or diskette, or in memory.

Remote Computer: A computing device external to the ERC Controller which has the capability to communicate with the ERC for purposes of information transfer.

Re transmit: To send a packet again if the original packet is not acknowledged or if it is received in error.

Retry: The process of retransmitting a block of data a prescribed number of times.

ROM (Read Only Memory): Memory whose contents are fixed during the manufacturing process, and that can not be changed by the user (except by physical replacement of the ROM).

RS-232C: The interface between DTE and DCE employing serial, binary data exchange.

RTS (Request To Send): An RS-232C control line between a modem and user equipment which initiates the data transmission sequence on a communications line.

Serial Transmission: A method of information transfer in which the bits of a character are sent serially in time (i.e. one at a time).

Shielding: A protective coating used on cables to reduce the interference and corruption of data which may be caused from external sources.

Simplex: The operation of a communication channel in one direction only, without the ability to transmit in the other direction.

Software: All programs which direct the function of the computer hardware.

SOH (Start Of Header): A control character signifying the start of the message heading data block.

Source: The origination of information or data transmission.

STX (Start Of Text): A control character signifying the end of the heading data and the start of information data.

Stop Bit: The last element of a character in asynchronous, serial transmission which is used to ensure the recognition of the next start element.

Storage: A device into which data can be entered, held, and retrieved at a later time.

Synchronous Transmission: A transmission process which utilizes a constant time interval between successive bits or characters.

Time-out: The expiration of a predefined time period, at which some specific action occurs.

Transaction: A message directed to an application.

Transmission Speed: The number of information elements sent per unit of time, usually expressed as bits, characters, or words. The preferred expression is bits per second (bps).

Volatile Storage: A storage device whose contents are lost when power is turned off or fails.

WACK (Wait Before Transmission): A code sent to a transmitting station indicating that the receiving station is not yet ready to receive additional data.

Write: To record data on a disk or diskette, or in RAM.

XON-XOFF: A flow control method which represents Transmitter On/Transmitter Off.

Notes

Notes